

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Leon Ropoša

**Ugotavljanje čustev na obrazu med
degustacijo hrane z metodami
računalniškega vida**

MAGISTRSKO DELO
ŠTUDIJSKI PROGRAM DRUGE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Franc Solina

Ljubljana, 2015

Rezultati magistrskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov magistrskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

IZJAVA O AVTORSTVU MAGISTRSKEGA DELA

Spodaj podpisani Leon Ropoša sem avtor magistrskega dela z naslovom:

Ugotavljanje čustev na obrazu med degustacijo hrane z metodami računalniškega vida

S svojim podpisom zagotavljam, da:

- sem magistrsko delo izdelal samostojno pod mentorstvom prof. dr. Franca Soline,
- so elektronska oblika magistrskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko magistrskega dela,
- soglašam z javno objavo elektronske oblike magistrskega dela v zbirki "Dela FRI".

V Ljubljani, 26. september 2015

Podpis avtorja:

Zahvaljujem se mentorju prof. dr. Francu Solini in somentorju viš. pred. dr. Borutu Batagelju za uso strokovno vodenje in nasvete pri izdelavi magistrskega dela.

Posebej se še zahvaljujem vsem, ki so si vzeli čas za pripravo baze posnetkov.

Kazalo

1	Uvod	1
2	Pregled uporabe metod za prepoznavo obraznega izraza na problemu okušanja hrane	5
3	Uporabljeni koraki v implementaciji	11
3.1	Detekcija obraza in obraznih značilk	11
3.2	Normalizacija in poravnava obraza	23
3.3	Izločanje značilk	30
3.4	Klasifikacija značilk	35
4	Rezultati na standardnih bazah	39
4.1	Opis baze Cohn Kanade CK+	39
4.2	Opis baze GEMEP-FERA	40
4.3	Izhodiščna metoda	41
4.4	Uporaba EAI predstavitve obraza	44
4.5	Preizkus učne množice obeh baz na posnetkih v realnem času	47
5	Ugotavljanje odziva na okušanje hrane in pijače na lastni bazi	51
5.1	Priprava in opis baze s posnetki okušanja	51
5.2	Prvotni rezultati na bazi	54
5.3	Nadaljni poizkusi in izboljšave	61

KAZALO

6	Iskanje konca prekrivanja obraza pri vnosu hrane in pijače	67
6.1	Iskanje z detektorjem obraznih značilk, ki ugotavlja ali so značilke prekrite	69
6.2	Iskanje z detekcijo sprememb v vektorskemu polju slike zaradi nastopa prekrivanja	81
7	Sklepne ugotovitve	89

Seznam uporabljenih kratic

kratica	angleško	slovensko
SIFT	scale invariate feature transformation	skalirno invariantna transformacija
LBP	local binary pattern	lokalni binarni vzorci
LPQ	local phase qunatization	lokalna kvantizacija faze
PCA	principal component analysis	metoda glavnih komponent
SVM	support vector machine	metoda podpornih vektorjev
FACS	facial action coding system	sistem predstavitve obraznih akcij
AU	action unit	enota akcije
FACS	facial action coding system	kodirni sistem obraznih akcij
CLM	constrained local model	omejeni lokalni model
HOG	histogram of oriented gradients	histogram orientiranih gradientov

Povzetek

V magistrski nalogi je predstavljena uporaba metod za prepoznavo obraznega izraza na problemu ugotavljanja ali je bil osebi okus zaužite hrane všeč ali ne. Trenutno obstaja nekaj raziskav, kjer so z uporabo obstoječe programske opreme, ki zazna prisotnosti šestih osnovnih čustev in nevtralnega izraza, iskali korelacijo med intenziteto zaznanih čustev in odzivom na okus hrane. V magistrski nalogi pa smo problem obravnavali kot dvorazredni, torej se na posnetku izraža prijeten ali neprijeten odziv na okus hrane. Na posnetkih odziva na okus hrane smo tudi učili uporabljeno metodo, medtem ko so v omenjenih raziskavah uporabili že naučene modele. Ločeno smo obravnavali odziv na okus hrane in odziv na okus pijače. Ker nismo našli nobene baze posnetkov okušanja, smo bazo pripravili sami. Najprej smo preizkusili nekaj metod za prepoznavo obraznega izraza na standardnih bazah, ki vsebujejo prikaz različnih čustev. Metodo, ki se je najbolj odrezala na bazah, smo potem uporabili na naši podatkovni bazi, pri čemer smo dosegli dobre rezultate. Na koncu smo še predstavili in preizkusili nekaj metod za avtomatsko ugotavljanje prekrivanja na obrazu, ki nastopi pri zajemu hrane. Pri preizkusu na naši podatkovni bazi smo namreč uporabili zgolj del posnetka, po koncu prekrivanja.

Ključne besede

Računalniški vid, prepoznavo obraznega izraza, detekcija čustev, ugotavljanje odziva na hrano, detekcija prekrivanja obraza

Abstract

Determination of Facial emotions during food testing using computer vision methods

This master thesis presents usage of facial expression methods for the purpose of discovering food taste reaction. There are currently a few research articles where using existing software that detects six basic emotions and neutral expression, correlation between intensity of detected emotions and food taste reaction was investigated. In this theses we presented this problem as binary, so that a video shows either pleasant or unpleasant food taste reaction. We also trained our method on this data, while the existing research used pretrained data. We separately tested response of reactions on food taste and drink taste. Because we could not obtain any database of food tasting, we prepared the database ourselves. We first tested some facial expression recognition methods on standard databases, that contain several different emotions. The most successful method was then used on our database, where it performed with good results. In the last chapter of thesis, we also presented and tested methods for automatic detection of face occlusion, which is the results of food intake. While performing test on our database we only used the part of video after the end of face occlusion.

Keywords

Computer vision, facial expression recognition, emotion detection, food taste reaction detection, face occlusion detection

Poglavje 1

Uvod

Metode za prepoznavo obraznih izrazov so prisotne že nekaj časa. Medtem ko dajejo dobre rezultate na standardnih podatkovnih bazah in v kontroliranih okoljih, se pri uporabi metod v realnem življenju pogosto srečamo s težavami. Za idealen sistem za prepoznavo obraznih izrazov naj bi veljalo [1]: da je neodvisen od značilnosti oseb kot so starosti, spol in rase posameznika, da je neodvisen od pogojev v okolju kot so osvetlitev ali pa prekrivanje obraza in da je popolnoma avtomatiziran. Večina raziskav na področju prepoznave obraznih izrazov z računalniškim vidom, poteka na detekciji tako imenovanih šestih osnovnih čustev: jeza, strah, gnus, presenečenje, veselje in žalost (Slika 1.1). Teh šest čustev je v 70. letih 20. stoletja psiholog Paul Ekman definiral kot univerzalna [1], to pomeni, da so prisotna v vseh kulturah, skozi celotno človeško zgodovino. V bazah in člankih se včasih pojavijo tudi katera dodatna čustva, pogosto se poleg teh šestih čustev doda tudi nevtralni izraz, ko ni zaznano nobeno čustvo.















Klasificiramo lahko direktno v čustvo, ali pa v predstavitev FACS (Facial Action Coding System), sistem za parametrizacijo obraznih izrazov, iz katerega potem razberemo čustva [1]. Sistem FACS izvira iz psihologije. Definira več enot akcije - AU(action units), ki nam povedo ali so na obrazu prisotni določeni gibi, kot so dvignjene obrvi, odprta usta(Slika 1.2). Vsaka izmed AU ima svoj identifikator, poleg tega k identifikatorju dodamo tudi opis in-



Slika 1.1: 6 osnovnih čustev kot jih je definiral Paul Ekman [3].

tenzivnosti izraza (s črkami od A do E). Klasifikacija v FACS je zahtevnejša, v magistrski nalogi se z njo nismo ukvarjali, saj bi za to potrebovali posnetke okušanja, ki bi bili kodirani s FACS. Pri klasifikaciji obraznih izrazov lahko upoštevamo časovne spremembe v posnetku, lahko pa obravnavamo zaporedne sličice ločeno, brez upoštevanja časovnih sprememb.

Poleg šestih osnovnih čustev se uporablja pri prepoznavi obraznih izrazov tudi prostor Prijetnost/Vzburjenje (ang. Valence/Arousal) [32]. V tem primeru prevedemo problem iz klasifikacijskega v regresijski, vzburjenje pomeni nivo intenzitete čustva. Možne so tudi zelo specifične aplikacije metod za prepoznavo obraznih izrazov, recimo za ugotavljanje utrujenosti voznikov ali pa za ugotavljanje bolečine iz obraznega izraza [31]. V takih primerih ne treniramo in klasificiramo na šestih osnovnih čustvih, temveč na naši problematiki - torej ali je na obrazu prisoten izraz bolečine. Alternativno lahko treniramo/klasificiramo s parametrizacijo FACS in iz konfiguracije AU razberemo ven prisotnost bolečine [31]. Problematiko magistrske naloge smo

AU1  Inner brow raiser	AU2  Outer brow raiser	AU4  Brow Lowerer	AU5  Upper lid raiser	AU6  Cheek raiser
AU7  Lid tighten	AU9  Nose wrinkle	AU12  Lip corner puller	AU15  Lip corner depressor	AU17  Chin raiser
AU23  Lip tighten	AU24  Lip presser	AU25  Lips part	AU27  Mouth stretch	

Slika 1.2: Prikaz posameznih AU katere definira FACS [2].

obravnavali kot binarni problem, torej se v posnetku izraža pozitiven ali negativen odziv na okus zaužite hrane.

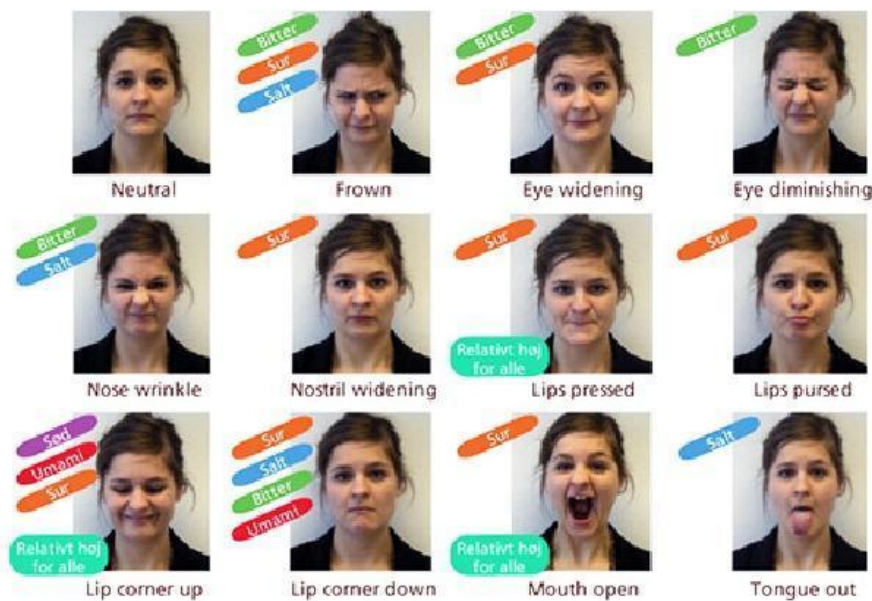
V 2. poglavju so najprej predstavljene dosedanje raziskave, kjer so za ugotavljanje odziva na okus hrane uporabili obstoječo programsko opremo, ki zazna šest osnovnih čustev. V 3. poglavju so opisani možni zaporedni koraki za metode, ki smo jih kasneje uporabili v 4. in 5. poglavju. V 4. poglavju smo preizkusili različne metode na standardnih bazah in na posnetkih prek spletne kamere. Metodo, ki se je najbolje obnesla v 4. poglavju, smo nato uporabili v 5. poglavju. V 5. poglavju je opisana priprava baze posnetkov in rezultati naših preizkusov na tej bazi. V 6. poglavju so opisani možni pristopi za odkrivanje prekrivanja, s katerimi bi določili začetek dela posnetka, ki bi se uporabil pri klasifikaciji.

Poglavje 2

Pregled uporabe metod za prepoznavo obraznega izraza na problemu okušanja hrane

Motivacija za izdelavo magistrske naloge je uporaba metod za prepoznavo obraznega izraza v promocijskih degustacijah hrane, z namenom ugotavljanja odziva kupcev na izdelek, predvsem v smislu prijetnosti okusa zaužite hrane. V praksi se ugotavljanje odziva meri ali z vprašalniki, pri čemer so lahko odgovori kupcev nezanesljivi, kupci pogosto nimajo volje izpolnjevati vprašalnika, ali pa s subjektivno presojo ljudi, ki na promocijah ponujajo hrano, ki je lahko tudi nezanesljiva. V tej situaciji bi lahko z natančnim sistemom za prepoznavo obraznega izraza dobili objektivno meritev.

V članku [17] so primerjali izraze na obrazu ljudi glede na okus zaužite hrane, v povezavi s prijetnostjo, kot tudi intenziteto okusa. Okuševalci so preizkušali vodne raztopine, ki so vsebovale enega izmed osnovnih okusov (sladko, grenko, kislo, slano in mesni okus) ali pa so bile brez okusa - navadna voda. Vodne raztopine so vsebovale različno intenziteto okusa, v splošnem je prijetnost okusa pri okuševalcih upadala z naraščanjem intenzitete okusa. Odkrili so povezave med določenimi okusi ter določenimi izrazi na obrazu, to je prikazano na sliki 2.1. Prav tako so se obrazni izrazi spreminjali z



Slika 2.1: Povezave med okusom hrane in obraznimi izrazi [18].

intenziteto okusa. Najbolj izraziti so bili obrazni izrazi pri visokih koncentracijah neprijetnih okusov, kot sta kislo (najbolj opazni so bili izrazi na ustnicah) in grenko (najbolj opazni so izrazi na očeh in čelu). Okusi, ki so bili okuševalcem najbolj prijetni (nizka in srednja intenziteta sladke raztopine in navadna voda), so sprožili na obrazu najmanj opazne obrazne izraze. To je v skladu z ugotovitvami prejšnjih raziskav, da so na obrazu najbolj izraziti predvsem okusi, ki jih zaznamo kot najmanj prijetne. V poizkusu so obrazne izraze tudi predstavili s sistemom FACS. Primer tega je viden na sliki 2.2.

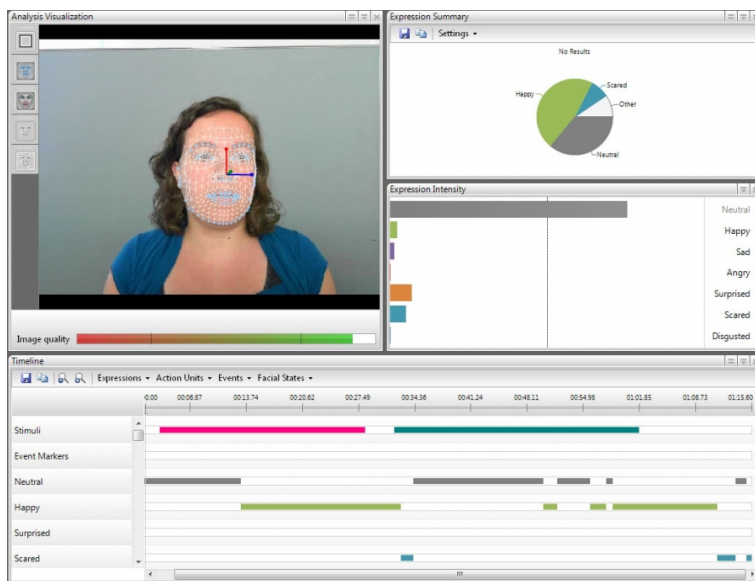
Obstaja nekaj člankov, kjer so z namenom ugotavljanja odzivov ljudi na zaužito hrano, uporabili metode za prepoznavo obraznega izraza. Trenutno ni na voljo nobenih baz s posnetki okušanja hrane, tako da so v člankih avtorji uporabili lastne posnetke okušanja. Avtorji člankov so sicer aktivni na področju raziskav povezanih s hrano, zato se članki ne osredotočajo na metode računalniška vida, temveč opisujejo kako so z uporabo obstoječe programske



Example of coding procedure

- AU 4* Brows are lowered and pulled together, evidenced by bulges on the forehead.
AU 9 The nose is wrinkled, seen by a bulge on the top of the bridge of the nose. Nose wrinkling also causes the nasolabial furrows to deepen.
AU 24 Lips are pressed.
AU 43 Eyes are closed to a maximum.

Slika 2.2: Določene AU enote na sliki odziva na okus pijače [19].



Slika 2.3: Delovanje programa Face Reader [22].

opreme analizirali odziv na zaužito hrano.

V vseh omenjenih člankih so za prepoznavo obraznih izrazov uporabili program FaceReader [21]. To je komercialni program katerega je razvilo podjetje Noldus. Program napove prisotnost enega izmed šestih osnovnih čustev in nevtralnega izraza. Za vsakega izmed šestih čustev in nevtralnega izraza, nam pove intenziteto čustva na obrazu. Program zazna tudi orientacijo glave in usmeritev pogleda. Klasifikacija v programu poteka s 3 nivojsko nevronske mreže, program sledi 500 ključnim točkam na obrazu. Delovanje programa prikazuje slika 2.3.

Članek [33] obravnava reakcije ljudi glede na prijetnost okusa sadnih sokov. V članku so uporabili sadne sokove, zato ker program FaceReader lahko interpretira gibe, ki nastanejo pri žvečenju hrane, kot obrazne izraze. V raziskavi so opravili dva ločena poizkusa, eksplicitni in implicitni poizkus. Razlika med njimi je bila, da so v eksplicitnem poizkusu okuševalci po zaužitju sokov, imeli 20 sekundni premor, po teh 20 sekundah pa so z ustreznim obraznim izrazom pokazali prejšnji odziv na okus soka. V impli-

citnem poizkusu uporabili zgolj posnetke po zaužitju sokov. Okuševalci so morali tudi na lestvici od 1 (najbolj všeč) do 9 (najmanj všeč) oceniti okus soka. S FaceReader-jem so po zaužitju sokov analizirali obrazne izraze na okuševalcih. Pri eksplicitnem poizkusu sta bili najbolj izraziti čustvi gnus in jeza, kadar je imel sok neprijeten okus in veselje ob prijetnem okusu. Pri implicitnem poizkusu sta bila gnus in jeza tudi prisotna, medtem ko veselje ni bilo več tako prisotno pri sadnih sokov s prijetnimi okusi. Končen sklep članka je, da so metode za prepoznavanje obraznega izraza, lahko primerne za ugotavljanje okusnosti pijače, pod pogojem da je okolje kontrolirano, kar sicer ni nujno pri nekem promocijskem mestu.

V članku [35] so merili različne telesne odzive na več vrst zaužite hrano, tako pri otrocih kot pri odraslih. Ena izmed meritev so bili tudi izrazi na obrazu. Pred začetkom poizkusa so okuševalci morali na vprašalniku iz skupine različnih jedi izbrati, katere so jim všeč in katere ne. Okuševalci so nato dobili v okušanje tri jedi, ki so jim bile všeč in tri jedi, ki jim niso bile všeč. Pri okušanju hrane so merili spremembe v intenziteti zaznanih čustev. Intenziteto čustev so merili v oknu na intervalu od 0.5 do 1.5 sekund pred in po okušanju. Razliko v spremembi intenzitete so določili kot največjo razliko v intervalu pred in po okušanju. Opazili so, da se je intenziteta zaznanih čustev spremenila predvsem pri neokusni hrani, pri okusni pa zgolj malenkostno. Spremembe v intenziteti tekom okušanja so bile zelo kratke, zaradi tega so avtorji sklepali, da bi merjenje obraznega izraza bilo uporabno zgolj pri kratkih okušanjih. Opazne so bile tudi spremembe na obraznih izrazih ob prvem vidnem stiku z okušano hrano. Avtorji članka so kasneje ponovili podoben poizkus še enkrat, le da so merili odziv na pijačo, natančneje pet vrst jogurtov in sadnih sokov [36]. Pri poizkusu so odziv na isto pijačo merili večkrat. Ugotovili so, da so povezave med obraznimi izrazi najbolj izrazite pri prvem okušanju pijače, in so z dodatnim okušanjem vse šibkejšje. Pozitivni odziv na okus pijače je imel največjo pozitivno korelacijo z nevtralnimi izrazom in negativno korelacijo z žalostjo, jezo, presenečenjem in strahom.

V članku [37] so merili odzive okuševalcev na različne vrste čokolad in

sladkarij. Okuševalci so prijetnost okusa navedli na lestvici od 1 (zelo neprijetno) do 7 (zelo prijetno). Poizkus so izvedli na podoben način kot so avtorji [33] izvedli ekspliciten poizkus, torej so okuševalci po zaužitju, morali z izrazom na obraz prikazati kakšen se jim je zdel okus hrane. Kot najbolj informativna čustva so našli veselje za pozitiven odziv in žalost za negativen odziv. Opazili so, da so določene vrste sladkarij in čokolade, dale v povprečju bolj izrazito intenziteto za določena čustva. Avtorji članka so ocenili da bi s FaceReader-jem lahko uspešno ocenili pozitivni in negativni odziv, glede na vrsto hrane, ki so jo obravnavali v članku.

V članku [34] so primerjali odziv na obrazu pri ljudeh z različnim indeksom telesne mase - BMI (Body Mass Index), pri zaužitju tekočine z grenkim okusom. Primerjali so razliko v intenziteti zaznanih obraznih izrazov 10 sekund pred zaužitjem in 10 sekund po zaužitju. Tudi tukaj so morali okuševalci navesti prijetnost okusa (na lestvici od 1 do 9, 1 najmanj všeč, 9 najbolj, 5 nevtrarno). Opazili so, da je bila sprememba intenzitet čustev povezana z oceno na lestvici. Nižja je bila ocena, večja je bila sprememba pri čustvu gnus, pri večji oceni je bila bolj izrazita sprememba v nevtralnem izrazu. Odkrili so, da je bila sprememba v intenziteti pri teh dveh čustvih, večja pri ljudeh s povišanim BMI.

V tem poglavju so predstavljene dosedanje raziskave na problemu uporabe prepoznavne obraznih izrazov pri določanju odziva na zaužito hrano. V 5. poglavju smo se problema lotili drugače kot v raziskavah, saj smo uporabljeno metodo učili na primerih odziva na hrano medtem, ko so v omenjenih raziskavah iskali korelacijo med vrsto odziva in zaznanimi intenzitetami za šest osnovnih čustev in nevtralni izraz. Pri pripravi baze posnetkov v 5. poglavju smo se zgledovali po tukaj omenjenih člankih. Naslednje poglavje opisuje možne korake v metodi, uporaba metod iz teh korakov je opisana v 4. in 5. poglavju.

Poglavje 3

Uporabljeni koraki v implementaciji

Prepoznavna obraznih izrazov z metodami računalniškega vida obsega naslednje zaporedje korakov: najprej detekcija obraza in po potrebi tudi detekcija lokacije določenih obraznih značilk na obrazu, nato poravnava zaznane slike obraza in normalizacija poravnane slike, predstavitev slike obraza v značilke primerne za klasifikacijo in na koncu klasifikacija z uporabo metod strojnega učenja.

3.1 Detekcija obraza in obraznih značilk

3.1.1 Detekcija obraza s Haarovimi značilkami

Metoda avtorjev Viola in Jonesa za detekcijo objektov v realnem času [4] se najpogosteje uporablja za detekcijo frontalnih obrazov. Ostala področja uporabe so detekcija delov obraza kot so oči in usta, profilni obrazi, logotipi, itd. Je binarni detektor, torej nam pove ali nek objekt na sliki prisoten ali ne. V fazi učenja, moramo algoritmu podati slike, kjer je iskani objekt na sliki prisoten - pozitivni primeri, kot tudi slike kjer iskani objekt ni prisoten - negativni primeri.

Algoritem temelji na Haarovih značilkah, to so značilke pravokotne oblike,

ki so sestavljene iz temnih in svetlih področij [4], slika 3.1 prikazuje te značilke na obrazu. Za lažje iskanje teh značilk si pomagamo z integralnimi slikami [4]. Integralna slika je predstavitev slike, kjer nam vrednost v posameznem slikovnem elementu pove vsoto v področju od zgornjega levega roba slike, do tega slikovnega elementa. Vsota intenzitet slikovnih elementov na nekem področju se izračuna po sledečem postopku. V integralni sliki pogledamo vrednost za vse štiri kote področja. Potem seštejemo vrednosti v zgornjem levem kotu in spodnjem desnem kotu, od te vsote odštejemo vrednosti v spodnjem levem in zgornjem desnem kotu. Celoten postopek je prikazan na sliki 3.2. V fazi učenja se uporabi algoritem strojnega učenja Adaboost [4]. Ta algoritem izmed vseh mogočih značilk, sestavi klasifikator, ki temelji na več šibkih klasifikatorjih, ti šibki klasifikatorji potem napovejo prisotnost, z najbolj obetavnimi značilkami. V fazi iskanja obrazov območje pregledujemo s kvadratnim poljem, ki ga povečujemo za določen faktor. Zato da algoritem prepozna območje slike kot obraz, mora v fazi iskanja obrazov, območje zadostiti vsem šibkim klasifikatorjem. Ko nek klasifikator zavrže območje, iščemo obraz na nekem drugem območju. Adaboost v fazi učenja izbere šibke klasifikatorje tako, da prvi klasifikator v vrsti zavrže, kar največ negativnih primerov, naslednji klasifikator potem zavrže negativne primere, katere je prvi klasifikator narobe napovedal kot pozitivne, postopek se ponavlja za celotno dolžino vrste.

Uporabili smo implementacijo algoritma iz odprtokodne knjižnice OpenCV [23]. Uporabili smo funkcijo *detectMultiScale*, pri tem smo uporabili enega izmed natreniranih detektorjev, detektorji so priloženi v OpenCV-ju v obliki xml datotek. Nastavitev `CASCADE_FIND_BIGGEST_OBJECT` uporabimo, če imamo v posnetku zgolj eno osebo, zato da kot rezultat dobimo zgolj največji objekt [10]. Pomembna parametra za natančnost in čas iskanja sta tudi *scaleFactor* in *minNeighbors*. *ScaleFactor* nam pove za koliko naj se povečuje območje za iskanje obrazov, manjši faktor lahko poveča natančnost vendar tudi upočasni algoritem [10]. *MinNeighbors* s povečanjem povzroči da obdržimo zgolj bolj zanesljive rezultate, s tem je tudi čas preiskovanja krajši [10].



Slika 3.1: Prikaz lokacij Haarovih značilnk na sliki obraza [11].

Original	Integral	Original	Integral																																																																																																				
<table><tr><td>5</td><td>2</td><td>3</td><td>4</td><td>1</td></tr><tr><td>1</td><td>5</td><td>4</td><td>2</td><td>3</td></tr><tr><td>2</td><td>2</td><td>1</td><td>3</td><td>4</td></tr><tr><td>3</td><td>5</td><td>6</td><td>4</td><td>5</td></tr><tr><td>4</td><td>1</td><td>3</td><td>2</td><td>6</td></tr></table>	5	2	3	4	1	1	5	4	2	3	2	2	1	3	4	3	5	6	4	5	4	1	3	2	6	<table><tr><td>5</td><td>7</td><td>10</td><td>14</td><td>15</td></tr><tr><td>6</td><td>13</td><td>20</td><td>26</td><td>30</td></tr><tr><td>8</td><td>17</td><td>25</td><td>34</td><td>42</td></tr><tr><td>11</td><td>25</td><td>39</td><td>52</td><td>65</td></tr><tr><td>15</td><td>30</td><td>47</td><td>62</td><td>81</td></tr></table>	5	7	10	14	15	6	13	20	26	30	8	17	25	34	42	11	25	39	52	65	15	30	47	62	81	<table><tr><td>5</td><td>2</td><td>3</td><td>4</td><td>1</td></tr><tr><td>1</td><td>5</td><td>4</td><td>2</td><td>3</td></tr><tr><td>2</td><td>2</td><td>1</td><td>3</td><td>4</td></tr><tr><td>3</td><td>5</td><td>6</td><td>4</td><td>5</td></tr><tr><td>4</td><td>1</td><td>3</td><td>2</td><td>6</td></tr></table>	5	2	3	4	1	1	5	4	2	3	2	2	1	3	4	3	5	6	4	5	4	1	3	2	6	<table><tr><td>5</td><td>7</td><td>10</td><td>14</td><td>15</td></tr><tr><td>6</td><td>13</td><td>20</td><td>26</td><td>30</td></tr><tr><td>8</td><td>17</td><td>25</td><td>34</td><td>42</td></tr><tr><td>11</td><td>25</td><td>39</td><td>52</td><td>65</td></tr><tr><td>15</td><td>30</td><td>47</td><td>62</td><td>81</td></tr></table>	5	7	10	14	15	6	13	20	26	30	8	17	25	34	42	11	25	39	52	65	15	30	47	62	81
5	2	3	4	1																																																																																																			
1	5	4	2	3																																																																																																			
2	2	1	3	4																																																																																																			
3	5	6	4	5																																																																																																			
4	1	3	2	6																																																																																																			
5	7	10	14	15																																																																																																			
6	13	20	26	30																																																																																																			
8	17	25	34	42																																																																																																			
11	25	39	52	65																																																																																																			
15	30	47	62	81																																																																																																			
5	2	3	4	1																																																																																																			
1	5	4	2	3																																																																																																			
2	2	1	3	4																																																																																																			
3	5	6	4	5																																																																																																			
4	1	3	2	6																																																																																																			
5	7	10	14	15																																																																																																			
6	13	20	26	30																																																																																																			
8	17	25	34	42																																																																																																			
11	25	39	52	65																																																																																																			
15	30	47	62	81																																																																																																			
$5 + 2 + 3 + 1 + 5 + 4 = 20$		$5 + 4 + 2 + 2 + 1 + 3 = 17$																																																																																																					
		$34 - 14 - 8 + 5 = 17$																																																																																																					

Slika 3.2: Računanje integralne slike, na levi strani prikaz kako izračunamo iz intenzitet slikovnih elementov slike integralno sliko, na desni strani prikaz kako iz integralne slike s 4 vrednostmi izračunamo vsoto intenzitet slikovnih elementov v nekem polju [12].

Funkciji lahko nastavimo parametre *minSize* in *maxSize*, najmanjša in največja možna velikost iskanega objekta, z njima lahko tudi zmanjšamo čas iskanja [10].

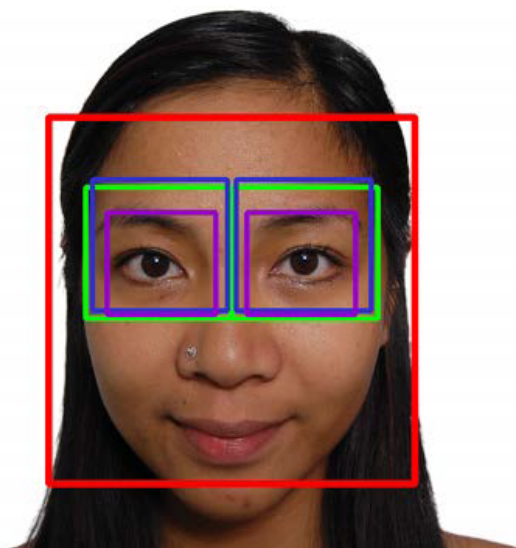
3.1.2 Detekcija oči s Haarovimi značilkami

Da smo lahko obraz poravnali smo morali na sliki obraza poiskati obe očesi. V primeru da lokacije obeh oces nismo našli, smo zaznano sličico obraza izpustili. Za detekcijo oči smo uporabili isti algoritem kot za detekcijo obraza iz knjižnice OpenCV, le da smo namesto xml datoteke za detekcijo obraza, uporabili eno izmed obstoječih xml datotek za detekcijo oči na podlagi Haarovih značilk. V [10] so za vsakega izmed detektorjev oči na podlagi Haarovih značilk opisane lastnosti kot so natančnost, hitrost in območje iskanja za največjo natančnost. Idealna območja iskanja so prikazana na sliki 3.3, ostale lastnosti pa v tabeli 3.1.

Tabela 3.1: Lastnosti Haar detektorjev za oči. Vsa imena detektorjev se začnejo s *haarcascade*.

Imena	Natančnost	Hitrost	Oči	Očala
<i>_mcs_lefteye.xml</i>	80%	18 ms	Odprte in zaprte	Ne
<i>_lefteye_2splits.xml</i>	60%	7 ms	Odprte in zaprte	Ne
<i>_eye.xml</i>	40%	5 ms	Samo odprte	Ne
<i>_eye_tree_eyeglasses.xml</i>	15%	10 ms	Samo odprte	Da

Pri nekaterih izmed detektorjev imamo dve datoteki, posebej za levo in za desno oko. Levo oko je v tem primeru levo oko osebe, torej bo na sliki obraza na desni strani. V primeru da bi uporabili detektor, ki je isti za levo in desno oko, bi ga uporabili dvakrat na obeh polovicah regijah oči. Za detekcijo oči smo uporabili *haarcascade_mcs_left(right)eye.xml*, zaradi največje natančnosti.



Slika 3.3: Prikaz idelanih območij iskanja za posamezne detektorje oči na sliki obraza [10]. Vijolična barva je za `eye.xml` in `eye_tree_eyeglasses.xml`, modra barva za `2splits.xml`, zelena barva pa za `mcs.xml`.

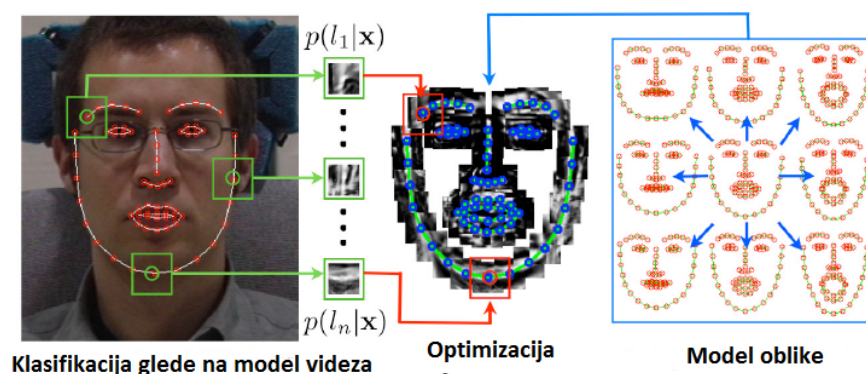
3.1.3 Omejeni lokalni model

Omejeni lokalni model (ang. Constrained Local model - CLM) [15], je ime za družino algoritmov za določanje lokacije obraznih značilnk. Zato da lahko določimo lokacije obraznih značilnk, najprej poiščemo obraz na sliki, recimo s prej omenjeno detekcijo obraza s Haarovimi značilkami, potem pa se na znani sliki obraza oceni lokacija obraznih značilnk, npr. robovi ust, robovi oči, robovi obraza. Na podlagi prejšnjih lokacij se potem tem obraznim značilkam lahko tudi sledi med trajanjem posnetka. Lokacije obraznih značilnk lahko uporabimo pri kasnejši poravnavi obraza. Kadar značilkam sledimo je postopek hitrejši od detekcije obraza s Haarovimi značilkami, saj ne preiskujemo vsakič celotne slike za lokacijo obraza.

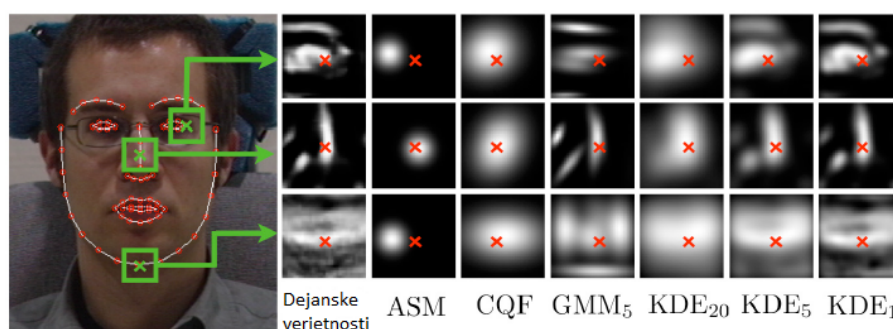
CLM temelji na rabi dveh modelov, modela videza obraznih značilnk (ang. patch model) [7], ki opisuje njihov izgled in modela oblike obraza (ang. shape model) [7], ki omejuje možne razporeditve obraznih značilnk. V fazi učenja moramo oba modela zgraditi iz učnih slik obrazov. Za učenje mo-

dela oblike obraza je primerna uporaba metode glavnih komponent - PCA (ang. Principal Component Analysis). Obliko obraza imamo v učnih podatkih, predstavljeno kot seznam točk z x in y koordinatama. Pred uporabo PCA moramo obraze v učnih podatkih tako poravnati, da iz njih odstranimo razlike zaradi translacij, rotacij in skaliranja. Zato da slike obrazov poravnamo uporabimo Procrustovo analizo [7]. Iz poravnanih slik obrazov potem izračunamo povprečno obliko obraza, torej obliko ki je najbolj skupna vsem obrazom. Povprečno obliko obraza potem odštejemo od vseh ostalih oblik v množici. Tako dobimo odklone različnih oblik obrazov od povprečja. Na teh izračunanih odklonih potem uporabimo PCA. Lastni vektorji, ki jih dobimo predstavljajo vrste odklonov od povprečne oblike (na primer širok ali ozek obraz), lastne vrednosti pa predstavljajo intenziteto odklonov. Zaradi uporabe PCA lahko potem vsako obliko obraza predstavimo kot linearno kombinacijo lastnih vektorjev s prišteto povprečno obliko obraza. Model videza nam prek klasifikatorja pove ali je območje vizualno podobno obrazni značilki ali ne, vsaka obrazna značilka ima svoj lastni klasifikator s posebej določenimi parametri. Klasifikatorju moramo podati negativne in pozitivne primere, vhodni podatki so lahko intenzitete slikovnih elementov na nekem območju, kot naprednejši, recimo izračunan vzorec LBP območja. V implementaciji, ki smo jo uporabili, je klasifikator logistična regresija [15]. Vhodni podatki, torej predstavitev območja, se najprej pošljejo OpenCV-jevi funkciji *matchTemplate*, ta funkcija v delu slike, ki ga preiskujemo, izračuna koliko je posamezno območje podobno iskani značilki. Te verjetnosti se potem uporabijo kot vhodni podatki za logistično regresijo, ta izračuna za vsako območje verjetnost, da to območje vizualno ustreza iskani obrazni značilki.

V fazi iskanja značilk potem upoštevamo tako model videza in model oblike. Najprej na zaznani sliki obraza določimo začetno oceno razporeditve lokacij obraznih značilk. Potem za vsako obrazno značilko preiščemo lokalno območje in dobimo verjetnosti za možne lokacije v območju. Izbrati moramo lokacijo obrazne značilke z največjo verjetnostjo, ki pa hkrati ni v nasprotju z modelom oblike. Optimizacijo glede na model oblike in model videza vidimo



Slika 3.4: Optimizacija glede na model videza po napovedanih verjetnosti iz klasifikacije območij (levo) in glede na možne razporeditve obraznih značil, katere definira model oblike (desno) [15].



Slika 3.5: Prikaz aproksimacije verjetnosti lokacije obraznih značil z uporabo različnih optimizacijskih strategij [15]. KDE (Kernel Density Estimation) je oznaka za našo uporabljeno strategijo. Uporabljena aproksimacija temelji na zgolj enemu parametru λ^2 , ki določa gladkost optimizacije, številka poleg oznake KDE prikazuje velikost parametra. Rdeči križec je dejanska lokacija obrazne značilke.

lahko na sliki 3.4. Preiskovanje za vsako obrazno značilko poteka neodvisno, torej je tukaj možnost pospešitve z uporabo paralelizacije. Preiskovanje lokacije obrazne značilke poteka večkrat zaporedoma, kot prekinitveni pogoj lahko omejimo število iteracij ali pa prenehamo z iskanjem, ko se lokacija obrazne značilke spreminja zgolj minimalno. Izkaže se, da je problem izbire najverjetnejše lokacije obrazne značilke, ki ni v nasprotju z modelom oblike, časovno zahteven. Naivna implementacija, kjer bi preiskali vsako možno lokacijo in izbrali tisto z največjo verjetnostjo, bi bila zelo počasna. Zato implementacije CLM uporabljajo druge strategije, ki verjetnosti iz klasifikatorjev videza, pretvorijo v posplešene predstavitve, ki jih lahko lažje rešimo. Primerjava kako različne metode aproksimirajo verjetnosti je vidna na sliki 3.5. Uporaba teh predstavitev je tudi manj občutljiva na obtičanje v lokalnem minimumu. Težave teh predstavitev so, da so odvisne od izbire večih parametrov in v mnogih primerih slabo aproksimirajo verjetnosti iz klasifikatorja. Naša uporabljena implementacija temelji na aproksimaciji z uporabo algoritma, podobnega algoritmu mean shift [16], ki se v računalniškem vidu sicer uporablja pri sledenju objektov. Postopek optimizacije v uporabljeni imple-

mentaciji CLM je prikazan s psevdokodo v algoritmu 1.

Algorithm 1: Omejeni lokalni model z mean shiftom omejenim v podprostoru (ang. Face Alignment through Subspace Constrained Mean-Shifts).

Imamo sliko I in lokacije obraznih značilk p .

```

while lokacija obraznih značilk  $p$ , ni skonvergirala do
    Za trenutna območja najdi verjetnostne porazdelitve z uporabo
    logistične regresije
     $x_i \approx x_i^c + J_i \delta p$ , model oblike se aproksimira s Taylorjevo
    razširitvijo prvega reda
    Izračuna se psevdo inverz Jacobijeve matrike  $J_i$  iz prejšnjega
    koraka, dobimo  $J_i^T$ 
     $\delta p = 0$ 
    while  $\delta p$  ni skonvergirala do
        Uporabi mean shift algoritem z Gausovim jedrom na
        porazdelitvi verjetnost, dobimo nove lokacij točk  $x_i^{(t+1)}$ 
         $\delta p = J^T [x_1^{(t+1)} - x_1^c; \dots; x_n^{(t+1)} - x_n^c]$ , na novih lokacijah točk
        se uporabi omejitev podprostora, določenega z modelom oblike
         $p = p + \delta p$ , popravi se lokacija obraznih značilk

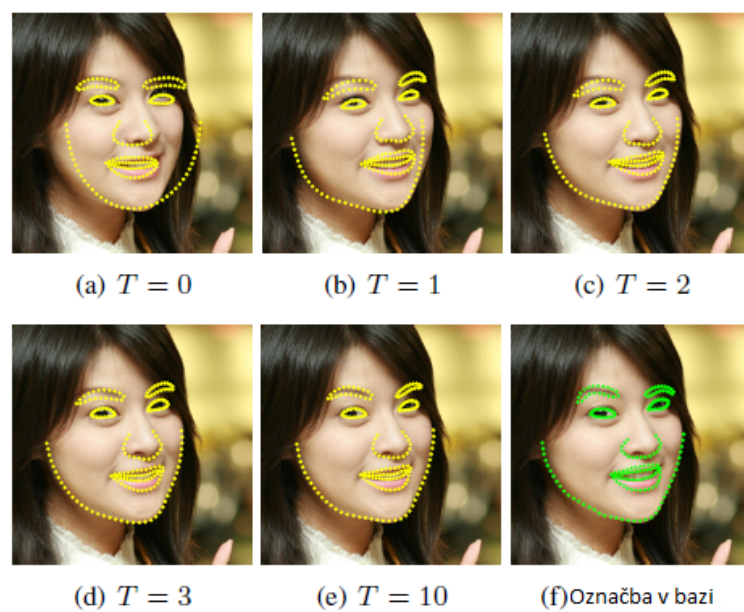
```

3.1.4 Določanje lokacije obraznih značilk z ansamblom regresijskih dreves

Metoda temelji na gradientnemu boostingu(ang. Gradient boosting) [26]. Gradientni boosting je algoritem strojnega učenja, soroden prej omenjenemu algoritmu Adaboost, saj imamo tukaj tudi vrsto šibkih klasifikatorjev, ki popravljajo napovedi prejšnjih klasifikatorjev. Pri Adaboostu se med posameznimi iteracijami spreminjajo uteži tako, da se povečujejo uteži za prej narobe napovedane primere in zmanjšujejo uteži za primeri, ki so bili pravilno napovedani. Tako se z učenjem skozi iteracije osredotočimo na težje primere v učni množici. Gradientni boosting se skozi iteracije uči na odvodu napake. Na začetku imamo za vse primere v učni množici enako napoved.

To napoved uporabimo kot učne podatke za prvi šibki klasifikator. Potem izračunamo negativni odvod funkcije napake za vse primere v učni množici glede na to napoved. Ta negativni odvod potem uporabimo kot resnične vrednosti za učne primere pri prvem šibkem klasifikatorju. Ko je prvi šibki klasifikator naučen, se potem napovedi, ki jih vrača, prišteje k dosedanjim napovedim za učne primere. Iz trenutnih napovedi se potem izračuna nov negativni odvod funkcije napake. Na naslednjemu šibkemu klasifikatorju se potem uporabi trenutno napoved kot učne podatke in negativni odvod kot resnične vrednosti za učne primere. Ta postopek se večkrat iterativno ponovi. Kot končno napoved se določi vsoto napovedi vseh šibkih klasifikatorjev. Ker gre v tem primeru za regresijski problem, saj napovedujemo lokacijo obraznih značilnk, se pri gradientnemu boostingu tukaj kot šibki model uporabi regresijsko drevo in negativni odvod kvadratne funkcije napake. Funkcija napake izračuna razlike posebej za vse pare značilnk, ki so prisotni na podatkih, na katerih učimo model. Začetno obliko obraza izračunamo tako, da izračunamo povprečno vrednost lokacije za vsako obrazno značilko v podatkih.

Kot kriterij za delitve v regresijskem drevesu, se vsakem vozlišču drevesa primerja, ali razlika intenzitet dveh slikovnih elementov presega določeno mejo. Če razlika presega mejo, nadaljujemo v levo poddrevo ali list, sicer pa v desno poddrevo ali list. Na začetku naključno izberemo določeno število lokacij slikovnih elementov znotraj začetnega obraza. Te lokacije se potem uporabijo tudi na nadaljnjih korakih algoritma. Natančneje uporabijo se lokacije, ki so enako oddaljene od trenutne razporeditve obraznih značilnk, kot so bile oddaljene začetne naključno izbrane točke od prvotne razporeditve obraznih značilnk. Zato moramo za vsako nadaljno obliko obraza izračunati transformacijo, ki preslika lokacijo točke iz začetne razporeditve obraznih značilnk v lokacijo točke, ki je enako oddaljena od trenutne razporeditve obraznih značilnk. Oddaljenost točke določimo kot oddaljenost začetne točke od njej najbližje obrazne značilke na začetni razporeditvi obraznih značilnk. Za preslikavo začetne oblike obraznih značilnk v trenutno obliko obraznih značilnk, moramo izračunati rotacijsko matriko in matriko skaliranja. Matriki sta



Slika 3.6: Prikaz delovanja vrste klasifikatorjev [26]. Na začetku obraz definirano kot povprečno vrednost obrazov, po 2 iteracijah gradientnega boostinga že imamo precejšnjo natančnost.

določeni tako, da je kvadratna razlika med trenutnimi lokacijami obraznih značilk in začetnimi lokacijami obraznih značilk pomnoženimi z matriko rotacije in matriko skaliranja, čim manjša. V fazi učenja, se ta transformacija izračuna za vsak obraz v vsaki iteraciji, v fazi napovedovanja pa zgolj enkrat za vsako iteracijo. V fazi napovedovanja je to tudi časovno najbolj potraten del algoritma.

Pri določanju kriterijev za delitve v regresijskemu drevesu, se pare točk in mejo za razlike njunih intenzitet določi naključno. Pri izbiri točk, se z večjo verjetnostjo izberejo točke, ki so bližje skupaj. Pri določanju kriterijev za delitev, začnemo v vozlišču drevesa in izberemo kombinacijo parov točk in meje razlike, ki učno množico razdeli tako, da je vsota kvadratnih napak za obe poddrevesi čim manjša. Postopek delitve lahko tudi ponovimo večkrat in potem izberemo najboljšega. V listih dreves določimo vektor, ki je enak povprečni vrednosti razlik negativnih odvodov, za vse učne primere v tem vozlišču. Primerom ki se znajdejo v tem listu, potem prištejemo ta vektor kot napoved. Gradientni boosting je občutljiv na preveliko prilagajanje učnim podatkov. Zato lahko, ko prištejemo napoved nove iteracije, to napoved pomnožimo s faktorjem omejitve (ang. shrinkage factor), ki leži med 0 in 1. Še boljši pristop regularizacije je, da v vsaki vrsti uporabimo večje število dreves in kot napoved posamezne iteracije, določimo povprečno napoved vseh dreves, število dreves med iteracijami je konstantno. Delovanje vrste pri napovedovanju lokacije obraznih značilk v posameznih iteracijah lahko vidimo na sliki 3.6.

Algoritem je implementiran v knjižnici dlib [24]. Dlib je odprtokodna knjižnica napisana v C++, ki vsebuje implementacije različnih algoritmov strojnega učenja, algoritme numeričnih metod in algoritme računalniškega vida. V knjižnici je priložen tudi naučen model, ki zazna lokacije 68 obraznih značilk. Slika 3.7 prikazuje delovanje implementacije iz knjižnice dlib. Poleg algoritma za detekcijo obraznih značilk, vsebuje knjižnica tudi algoritem za detekcijo obraza s priloženim naučenim modelom. Algoritem temelji na rabi značilk histograma orientiranih gradientov - HOG (Histogram of



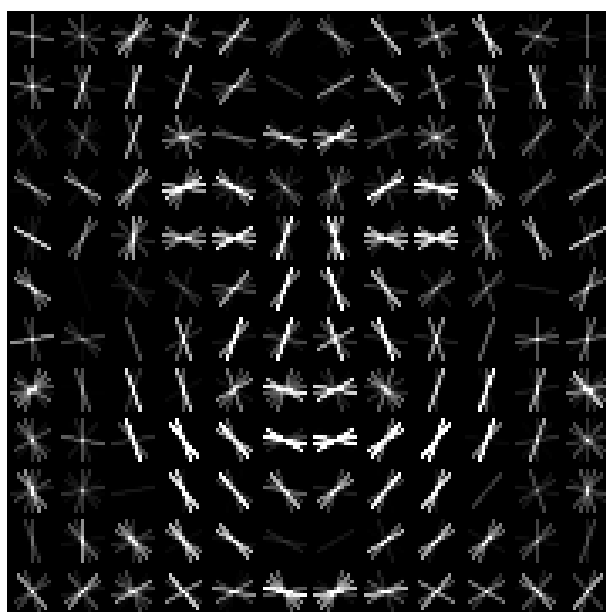
Slika 3.7: Prikaz delovanja algoritma za detekcijo obraznih značilk iz knjižnice dlib prek spletne kamere.

oriented gradients) [28], drsečega okna in linerarnih klasifikatorjih. Vizualizacija modela je prikazana na sliki 3.8. Algoritem je sicer možno uporabiti tudi za detekcijo drugih objektov. Detekcija obraza, kot tudi iskanje lokacije obraznih značilk, delujeta v realnem času. Največja razilka s prej omenjenim CLM je, da metoda implementirana v knjižnici dlib vsakič posebej zazna obraz in potem na obrazu poišče značilke, torej ne sledi značilkam med potekom posnetka glede na prejšnje lokacije. Knjižnico dlib je mogoče uporabiti istočasno skupaj z knjižnico OpenCV, prav tako je do knjižnice dlib mogoče dostopati prek Pythona ali Matlaba.

3.2 Normalizacija in poravnava obraza

3.2.1 Poravnava na zaznane oči

Postopek smo implementirali po zgledu iz [10]. Kadar smo obrazne značilke poiskali s pomočjo metode CLM ali s pomočjo knjižnice dlib, smo lokacijo centra oči zračunali kot povprečje lokacij obraznih značilk okoli posameznega očesa. Poravnava se določi na podlagi treh parametrov: velikosti končne slike (širina in višina slike sta enaka) in relativne lokacija centra levega očesa (ločena parametra za širino in višino). Postopek se izvrši z eno transformacijo z uporabo OpenCV funkcije *warpAffine*. Cilj transformacije je, da sta centra oči na koncu na isti višini in da je razdalja med njimi zmeraj enaka. Glede na



Slika 3.8: Vizualizacija naučenega HOG modela obraza z uporabo knjižnice dlib, v sredini lahko vidimo obris obraza [25]. Funkcija za vizualizacijo je del knjižnice dlib.

izbiri vrednosti relativne lokacije centra oči, lahko dosežemo tudi obrezovanje ozadja ali dela obraza, ki ga ne potrebujemo. Smiselne vrednosti za relativno lokacijo očesa, so med 0.14 in 0.36. Levo oko bo po transformaciji oddaljeno toliko od levega roba slike, kot bo desno oko od desnega. Zato da lahko funkcija *warpAffine* izvrši transformacijo, potrebuje transformacijsko matriko. To dobimo z uporabo OpenCV funkcije *getRotationMatrix2D*, funkcija kot parametre sprejme center rotacije, kot rotacije in nivo skaliranja. Center rotacije se določi kot središčna točka med centrom levega in desnega očesa. Kot rotacije določimo z uporabo funkcije *atan2*, kot parametra ji podamo, razliko med višino desnega in levega očesa in razliko med širino desnega in levega očesa. Nivo skaliranja določimo tako, da končno oddaljenostjo med levim in desnim očesom delimo s trenutno oddaljenost med očesi. Končno oddaljenost lahko določimo, tako, da pomnožimo razliko med relativnimi širinami desnega in levega očesa s širino končne slike. Dobljena rotacijska matrika ima 2 vrstici in 3 stolpce. Zato da dobimo željeno transformacijsko matriko, je treba v rotacijsko matriko dodati še translacijo ki bo center oči premaknila na pravo mesto. Prvi vrstici tretjega stolpca se prišteje razliko med polovico končne širine slike in trenutno širino centra oči. Drugi vrstici tretjega stolpca se prišteje premik centra oči na pravo višino. Dobimo ga tako, da relativno višino levega očesa pomnožimo z višino končne slike obraza in od tega odštejemo trenutno višino centra oči. V magistrsko nalogi smo kot relativne lokacija levega centra oči uporabili 0.24 za x koordinato in 0.20 za y koordinato. Vpliv izbire različnih vrednosti je viden na sliki 3.9. Celoten

postopek poravnave je tudi prikazan s psevdoko v 2.

Algorithm 2: Poravnava obraza glede na zaznan center oči.

```

Podani parametri: velikost_obraza(širina in dolžina sta ista), relativni
lokaciji levega očesa(rel_left_eye z koordinatama  $x$  in  $y$ )
rel_right_eye.y = rel_left_eye.y
rel_right_eye.x = 1 - rel_left_eye.x
konca_razdalja_med_ocesi = (rel_right_eye.x - rel_left_eye.x) *
velikost_obraza
center_oci = Point( (rel_left_eye.x + rel_right_eye.x) * 0.5,
(rel_left_eye.y + rel_right_eye.y) * 0.5 )
// 2D točka z z koordinatama  $x$  in  $y$ 
dx = rel_left_eye.x - rel_right_eye.x
dy = rel_left_eye.y - rel_right_eye.y
razdalja_med_ocesi =  $\sqrt{(dx * dx) + (dy * dy)}$ 
kot_med_ocesi = atan2(dy, dx)
faktor_skaliranja = konca_razdalja_med_ocesi / razdalja_med_ocesi
rotacijska_matrika = getRotationMatrix2D(center_oci, kot_med_ocesi,
faktor_skaliranja)
rotacijska_matrika(0, 2) += velikost_obraza * 0.5 - center_oci.x
// prištejemo prvi vrstici in tretjemu stolpcu rotacijske matrike
rotacijska_matrika(1, 2) += velikost_obraza * rel_left_eye.y -
center_oci.y
// prištejemo drugi vrstici in tretjemu stolpcu rotacijske matrike
warpAffine(zaznan_obraz, poravnan_obraz, rotacijska_matrika)

```

3.2.2 Normalizacija histograma

Na sliki obraza smo po poravnavi na zaznane oči uporabili še normalizacijo histograma z uporabo OpenCV funkcije *equalizeHist*. Normalizacija histograma se uporablja za izboljšanje kontrasta in osvetljenosti slike. Da normalizacija to doseže spreminja histogram slike tako, da ga čimbolj izravna in ima histogram slike potem čimbolj enakomerno razporeditev intenzitet. Pri



Slika 3.9: Primerjava končnih slik, na isti vhodni sliki glede na relativno lokacijo levega centra oči. Zgoraj levo: $x=0.18$ in $y=0.16$, zgoraj desno $x=0.24$ in $y=0.20$, spodaj levo: $x=0.28$ in $y=0.24$, spodaj desno $x=0.32$ in $y=0.28$.



Slika 3.10: Zgoraj: levo slika obraza po transformaciji z neenakomerno osvetlitvijo po polovicah obraza, desno rezultat ločene normalizacije obraza po levi in desni strani. Spodaj: levo rezultat normalizacije obraza na celotni sliki, razlika med osvetlitvami polovic je še bolj vidna, desno končni rezultat ko združimo 3 normalizirane slike.

obraznih slikah z normalizacijo obraza zmanjšamo vpliv razlik v osvetlitvi okolja na končni sliki. Pri določenih osvetlitvah je lahko ena stran obraza bolj osvetljena kot druga stran. Zato smo po zgledu [10] izvedli ločeno normalizacijo obraza po levi in desni polovici obraza ter potem ti sliki združili. Slaba lastnost tega pristopa je, da lahko povzroči oster prehod med levo in desno polovico obraza.

Za bolj gladek prehod smo pri združitvi uporabili še sliko celotnega obraza na kateri smo izvedli normalizacijo [10]. Tako potem uporabimo tri slike pri združevanju v končno sliko. Postopek združevanja poteka tako, da obraz razdelimo na štiri enako velike dele po širini slike obraza. V prvi četrtini prepisemo zgolj intenziteto slikovnega elementa na istem mestu na normalizirani levi strani obraza. Podobno naredimo tudi v zadnji četrtini obraza,

le da tukaj uporabimo desno polovico obraza. Pri zadnjih dveh polovicah obraza moramo paziti, da prepisemo slikovne elemente iz desnega dela na končno sliko obraza z zamikom, ki vključuje širino levega dela obraza. Pri drugi in tretji četrtini obraza pa uporabimo tudi intenziteto celotnega normaliziranega obraza. Pri drugi četrtini povprečimo intenzitete na istem mestu na levi polovici in celotnemu normaliziranemu obrazu, pri tretji četrtini pa povprečimo celotni normaliziran obraz z desnim delom obraza. Povprečenje ni enakomerno, bolj se približujemo sredini obraza večjo težo dajemo intenziteti na slikovnem elementu na celotni normalizirani sliki obraza, bolj smo oddaljeni od sredine večjo težo damo intenziteti slikovnega elementa na levi oz. desni polovici obraza. Primer kako izgledajo različne normalizirane slike v postopku je viden na sliki 3.10.

3.2.3 Predstavitev algoritma EAI z uporabo Sift Toka

EAI (Emotion Avatar Image) [14] je algoritem, ki se je uporabil za poravnavo slike, pri zmagovalni metodi na tekmovanju GEMEP-FERA 2011 [27]. Algoritem za vsakega izmed posnetkov ustvari 1 sliko, ki predstavlja celotni posnetek. Posledično je metoda primerna kadar se v posnetku pojavi prikaz zgolj enega čustva, kot je to pri bazi GEMEP-FERA in manj primerna pri bazi Cohn Kanade CK+ ko iz nevtrane izraza preidemo v čustvo. Metoda temelji na uporabi algoritma Sift tok (ang. Sift Flow) [29]. Sift tok je podoben optičnemu toku, s to razliko da tukaj iščemo polje vektorjev, ki bo, namesto razdalje slikovnih elementov z isto intenziteto, minimiziralo SIFT značilke. SIFT (Scale Invariant Feature Transform) je algoritem ki najde ključne točke, za katere velja, da so značilke ki jih opisujejo, neodvisne od skaliranja, rotiranja in dodajanja šuma v slike. Sift tok uporablja tudi veliko večje okno kot optični tok. Formula, ki jo, minimiziramo je prikazana v enačbi 3.1. Prva vsota omejuje značilke Sift glede na tok vektorja, druga vsota omejuje vektorje zamika na čim manjšo vrednost (ang. small displacement term), tretja vsota omejuje tok vektorja na gladkost spremembe toka (ang. smoothness term) [29].

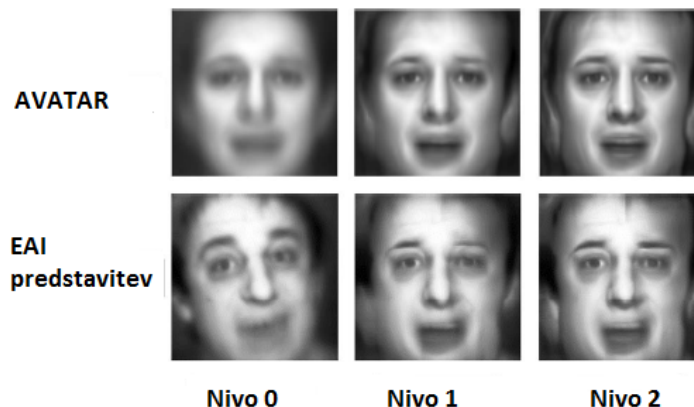
$$\begin{aligned}
E(w) = & \sum_p \min(\|s_1(p) - s_2(p + w(p))\|_1, t) \\
& + \sum_p \eta(|u(p)| + |v(p)|) \\
& + \sum_{(p,q) \in \epsilon} \min(\alpha|u(p) - u(q)|, d) + \min(\alpha|v(p) - v(q)|, d) \quad (3.1)
\end{aligned}$$

EAI predstavitev se izvede tako da se poravna slika obraza glede na referenčno sliko - Avatar [14]. Avatar je slika ki je zgrajena z povprečenjem poravnav preko Sift toka na prejšnjem nivoju. Začnemo z nivojem 0. Tukaj je Avatar slika povprečje vseh slik obrazov v učni množici. S tem ko povprečimo slike za posamezen posnetek dobimo predstavitev EAI posnetka na nivoju 0. EAI predstavitve se uporabljajo za pridobivanje značilnk za kasnejšo klasifikacijo. Potem vsako zaznano sliko obraza, poravnamo glede na Sift tok vektor, z referenco na sliko Avatarja. Za vsak posamezen posnetek potem povprečimo slike, ki smo jih prej poravnali. Povprečenje posnetka se izvrši tako, da seštejemo vse slike za posnetek in jih delimo z njihovim številom. Tako smo dobili EAI predstavitev posnetka na 1 nivoju. Če povprečimo vse EAI predstavitve posnetka na 1 nivoju dobimo, potem Avatar sliko na 1 nivoju. Če ponovimo postopek še enkrat bomo dobili Avatar sliko na 2 nivoju in pred tem EAI predstavitve na 2 nivoju. Po zgledu metode smo uporabili EAI predstavitve na 0, 1 in 2 nivoju. Primeri EAI predstavitev posnetka in avatarjev po nivojih so prikazani na sliki 3.11.

3.3 Izločanje značilnk

3.3.1 Lokalni binarni vzorci (LBP)

Lokalni binarni vzorci(ang. Local Binary Patterns) so način za opis teksture [5]. Vsak slikovni element na sliki opišemo z bitno binarno kodo, iz teh kod sestavimo histogram, ki pove število pojavitev posameznih kod. LBP

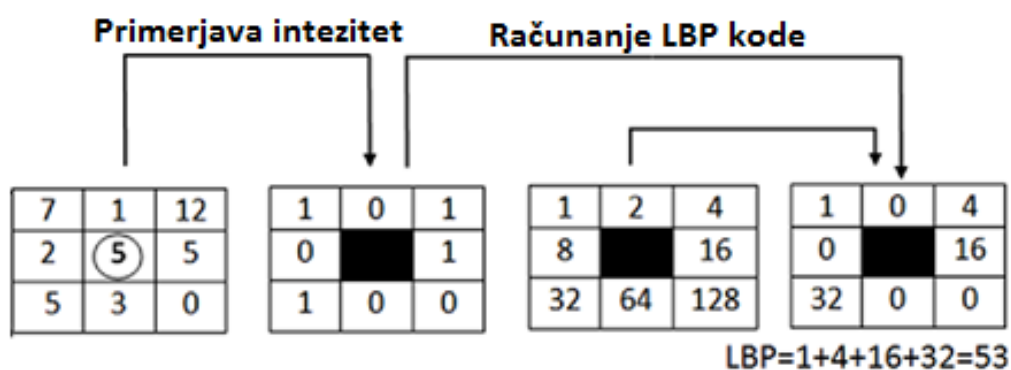


Slika 3.11: Primerjava slik Avatarja in EAI predstavitev na posameznih nivojih [14].

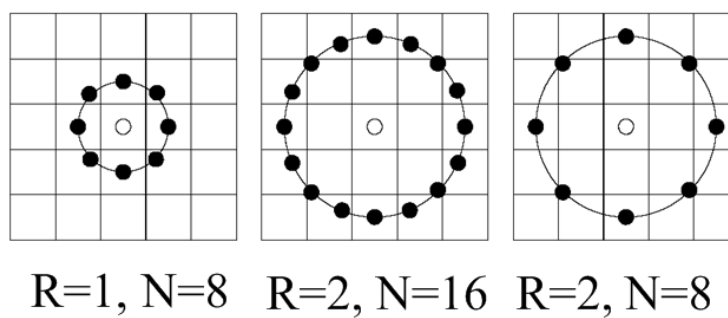
kodo določimo tako, da primerjamo intenziteto slikovnega elementa z intenzitetami slikovnih elementov v soseščini. Kadar je intenziteta slikovnega elementa v soseščini manjša od intenzitete slikovnega elementa v središču, napišemo na določeno mesto v binarno kodo 0, sicer pa 1. Vsak slikovni element v soseščini vpliva zgolj na vrednost enega izmed bitov kode, vsak bit kode je odvisen zgolj od enega slikovnega elementa. Postopek računanja vzorca je prikazan na sliki 3.12.

Soseščino, kjer računamo LBP kodo, se določi z dvema parametroma N - število slikovnih elementov s katerimi primerjamo središčni slikovni element, parameter določa tudi dolžino kode, ter R - radij kako daleč od središčnega slikovnega elementa vzamemo soseščino. V magistrski nalogi smo uporabili $N=8$ in $R=1$ [27]. S tema vrednostima parametrov je število možnih vzorcev 256. Primeri različnih soseščin so vidni na sliki 3.13.

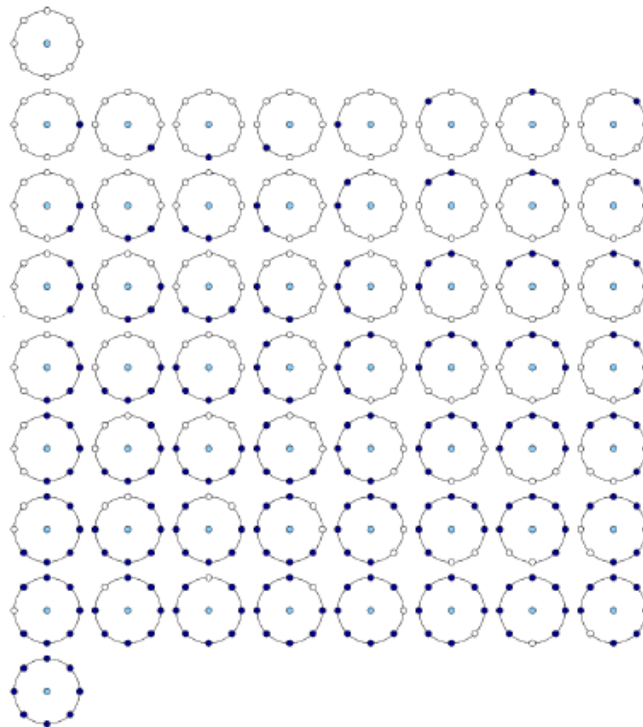
V praksi se pokaže, da se nekateri vzorci pojavljajo veliko bolj pogosto od ostalih, iz tega izvira nadgradnja operatorja LBP, uniformni lokalni binarni vzorci (ang. Uniform Local Binary Patterns) [5]. Za kode vzorcev, ki se pojavljajo pogosteje od ostalih velja, da v njih ni prehodov med ničlami



Slika 3.12: Računanje LBP vzorca [13].



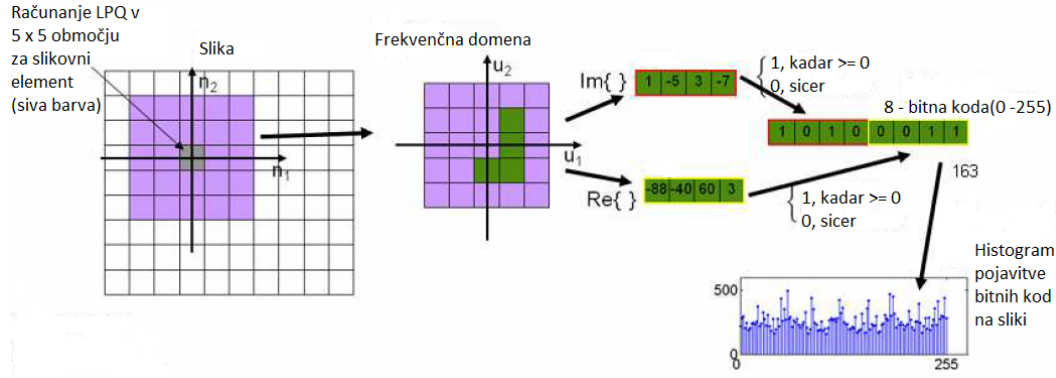
Slika 3.13: Različne sosesčine za vzorce LBP, prvi in tretji vzorec imata enako število elementov, drugi in tretji vzorec imata enak radij [13].



Slika 3.14: Vseh 58 uniformnih lokalnih binarnih vzorcev [6].

in enicami (torej je celotna koda same ničle oz. enice, dva vzorca) ali pa se pojavita dva prehoda med ničlami in enicami v kodi (en prehod ni možen, saj kodo gledamo ciklično). Pri izbiri parametrov $N=8$ in $R=1$, je uniformnih vzorcev 58, možne uniformne vzorce vidimo na sliki 3.14. Za preostale vzorce se določi ista koda, tako da na koncu dobimo 59 možnih kod. Prednost rabe uniformnih vzorcev je zmanjšanje števila značilk, kot tudi povečana robustnost operatorja na rotacije. Uniformne lokalne binarne vzorce označimo z oznako U2, tako da je oznaka uporabljenega operatorja LBP (8,1) U2.

Zato da ohranimo prostorske informacije o teksturi, sliko obraza razdelimo v mrežo, mi smo uporabili mrežo velikosti 10×10 [27]. Na vsakemu izmed 100 kosov mreže obraza ločeno izračunamo LBP histogram, vse te histograme potem združimo v enotni vektor značilk. V primeru operatorja LBP (8,1) U2 je potem dolžina vektorja za posamezno sliko obraza $10 * 10 * 59 = 5900$ [27].



Slika 3.15: Prikaz izračuna LPQ značilnk [6].

3.3.2 Lokalna kvantizacija faze (LPQ)

Lokalna kvantizacija faze (ang. Local Phase Quantization) [30], je nadgradnja algoritma LBP. Prvotni namen uporabe LPQ je za opis in klasifikacijo tekstur z zamegljivostjo (ang. blur). Pri teksturah z zamegljivostjo, da metoda boljše rezultate, kot LBP. V preizkusih avtorjev je metoda dala tudi malce boljše rezultate napram LBP, kadar na sliki ni prisotna zameglitev. Za vsak slikovni element izračunamo fazo z uporabo kratotrajne Fourierjeve transformacije (ang. short-term Fourier transform). V računanju se uporabijo 4 frekvence $u1=[a,0]$, $u2=[a,a]$, $u3=[a,a]$ in $u4=[a,-a]$, kjer je a nek parameter, uporabili smo $a = 1/7$. Vsaka izmed uporabljenih frekvenc da število z realnim in imaginarnim delom. Za vsako izmed dobljenih števil potem binariziramo tako realni kot imaginarni del, 1 če je število pozitivno in 0 kadar je negativno [30]. Postopek računanja vzorca LPQ je prikazan na sliki 3.15. Na koncu dobimo za posamezen slikovni element 8-bitno kodo, torej je 256 možnih različnih kod. Kot pri LBP, smo tudi pri LPQ sliko obraza razdelili v mrežo velikosti 10×10 [14]. Na vsakemu od teh kosov izračunamo histogram, tako da imamo na koncu vektor značilnk dolžine $10 * 10 * 256 = 25600$.

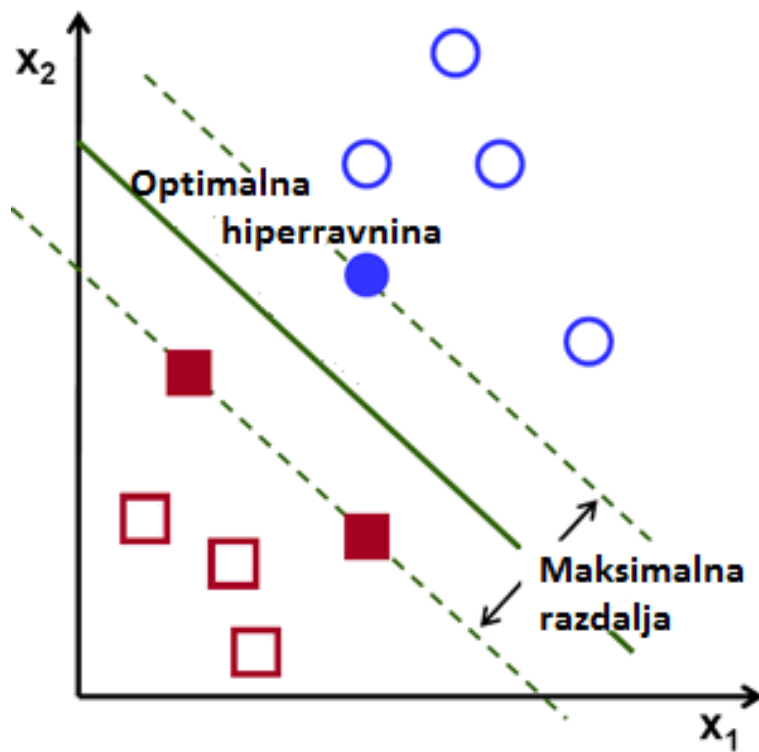
3.3.3 Metoda glavnih komponent (PCA)

Metoda glavnih komponent (ang. Principal Components Analysis) je statistični postopek, ki z uporabo ortogonalne transformacije, spremeni množico spremenljivk med katerimi obstaja možnost korelacije, v množico linearno neodvisnih spremenljivk. Tej množici rečemo tudi glavne komponente (ang. principal components). Značilnost teh komponent je, da ima prva komponenta največjo varianco - torej zaobjame kar največ odklona v podatkih, vsaka naslednja komponenta ima varianco omejeno navzgor z varianco prejšnje komponente. Če obdržimo določeno število glavnih komponent od začetka, dobimo manjše število spremenljivk z zanemarljivo izgubo informacije, saj preostale komponente vključujejo večino variacije v podatkih. Pogosta izbira je, da ohranimo tako število glavnih komponent, ki nam pustijo 95% variacije v podatkih. PCA uporabimo, ko smo z nekim drugim operatorjem že pridobili značilke, za zmanjšanje dimenzionalnosti podatkov, na katerih bomo vršili klasifikacijo. Zaradi manjše dimenzionalnosti podatkov, je kasnejša klasifikacija hitrejša, tako na testni, kot tudi na učni množici.

3.4 Klasifikacija značilk

3.4.1 Metoda podpornih vektorjev (SVM)

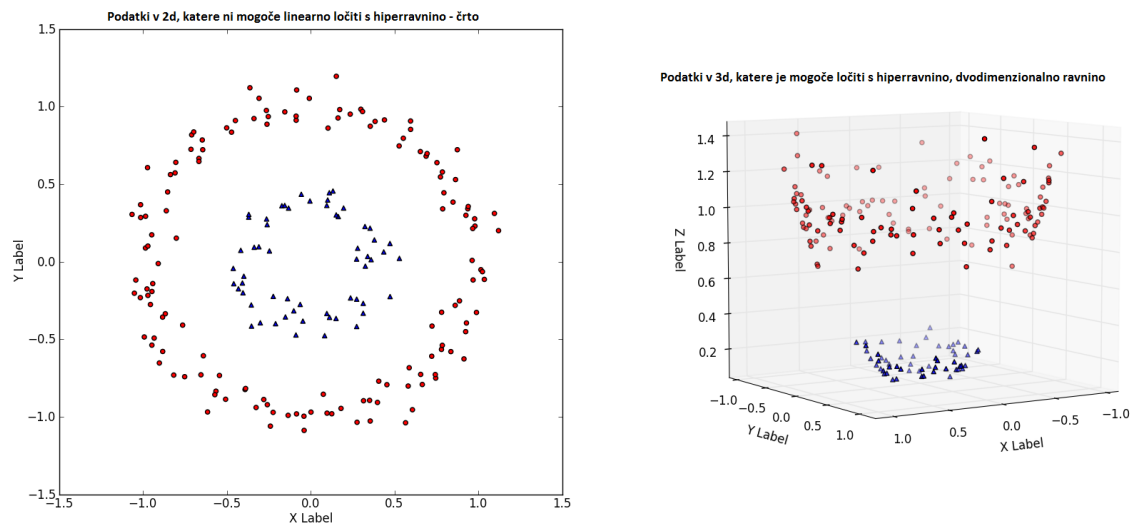
Metoda podpornih vektorjev (ang. Support vector machines) je ena izmed najbolj uporabljenih metod strojnega učenja. Metoda ustvari hiperravnino v prostoru značilk in s tem razdeli prostor med dva razreda. Hiperravnina je lik, ki je za eno dimenzijo manjši od prostora značilk, torej če imamo dvo-dimenzionalni prostor, bo hiperravnina črta, če imamo tro-dimenzionalni prostor bo hiper ravnina, ravna površina, itd... Ker je lahko za nek problem možnih več hiperravnin, ki ločijo prostor, se izmed njih izbere tista, ki maksimizira razdaljo učnih primerov od hiperravnine (ang. margin), ideja je prikazana na sliki 3.16. Kljub temu, da smo izbrali hiperravnino z največjim odklonom, obstaja možnost, da je bilo nekaj primerov narobe klasificiranih.



Slika 3.16: Osnovna ideja SVM je iskanje optimalne hiperravnine, ki ustvari največjo možno razdaljo učnih primerov od hiperravnine [8].

Z uporabo parametra C - konstante mehkega roba, SVM-ju določimo kazni, za katero naj se izogne napačni klasifikaciji. Ob primerni vrednosti C lahko SVM vrne hiperravnino z manjšim skupnim odklonom, ki pa narobe klasificira manj primerov.

Osnovna verzija SVM uporablja linearno jedro (ang. linear kernel) in deluje, kadar je podatke možno linearno ločiti. Kadar podatkov ni mogoče linearno ločiti, moramo uporabiti SVM z drugim jedrom. Kadar uporabimo tako jedro, podatke glede na funkcijo, ki ga to jedro definira, preslikamo v višje dimenzionalni prostor, kjer obstaja hiperravnina, ki uspešno razdeli prostor podatkov. Tako SVM naučimo v tem višje dimenzionalnem prostoru in ko hiperravnino najdemo, jo preslikamo nazaj v prvotni prostor in to je potem naša meja med razredi. Ideja uporabe SVM z jedrom je prikazana na



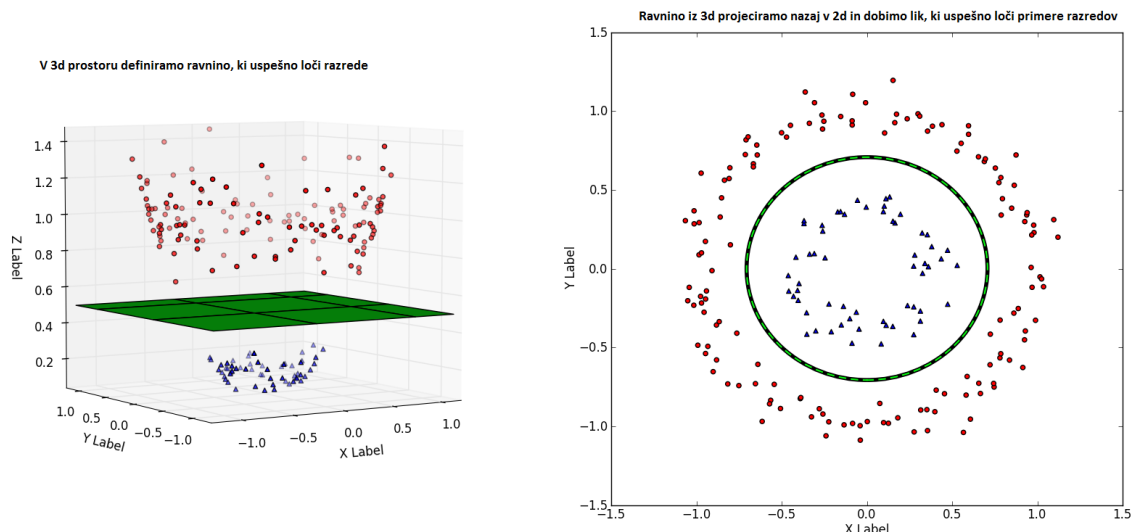
Slika 3.17: Uporaba jedra za transformacijo podatkov, ki jih ni mogoče linearno ločiti, v višjo dimenzijo [9].

slikah 3.17 in 3.18.

OpenCV-jeva implementacija SVM temelji na knjižnici LIBSVM. Kadar imamo problem klasifikacije z N različnimi razredi (N je večji kot 2), recimo napoved več različnih čustev, uporabimo eno izmed naslednjih strategij.

Pri strategiji "eden proti ostalim" (ang. one vs all) ustvarimo N klasifikatorjev, enega za vsako čustvo. Vsak klasifikator napove prisotnost enega čustva, to je pozitivni razred, za negativni razred - kadar čustvo ni prisotno, pa se uporabijo vsa ostala čustva v učni množici. Pri rabi te strategije, se lahko zgodi, da za posamezno sliko obraza dobimo, glede na rezultate klasifikatorjev, napovedano prisotnost večih čustev. Možno je tudi, da noben od klasifikatorjev ne zazna čustva, torej se čustvo ni pojavilo.

Pri strategiji "vsak z vsakim" (ang. one vs all) ustvarimo $(N*(N-1))/2$ klasifikatorjev za N različnih čustev. Pri tej strategiji ustvarimo klasifikator za vsaki možen par čustev, vsak klasifikator torej napove zgolj enega izmed dveh razredov. Na koncu izberemo kot zaznano čustvo tisto, ki je bilo največkrat napovedano. To strategijo lahko uporabimo, kadar smo prepričani, da se v



Slika 3.18: Končna hiperravnina z uporabo jedra [9].

posamezni sličici zagotovo pojavlja eno izmed čustev, sicer moramo dodati še nov razred, ki predstavlja, da nobeno izmed čustev ni prisotno.

Izmed jeder SVM klasifikatorja, ki so na voljo v OpenCV, smo uporabili jedro RBF (Radial basis function) in v primerih, ko je bilo število značilk znatno večje od števila učnih primerov, linearno jedro. Postopek prečnega preverjanja smo izvršili z uporabo metode `CvSVM::train_auto`, ki je vključena v OpenCV.

V tem poglavju smo predstavili možne korake iz katerih je sestavljena metoda za prepoznavo obraznega izraza. Tukaj opisane korake smo izbrali zato, ker so bili že v prejšnjih raziskavah uspešno uporabljeni na problemu prepoznave obraznih izrazov. Naslednje poglavje opisuje rezultate preizkusov metod na standardnih bazah.

Poglavje 4

Rezultati na standardnih bazah

4.1 Opis baze Cohn Kanade CK+

Baza Cohn Kanade CK+ [20] vsebuje 327 posnetkov v katerih nastopa 118 oseb, od tega 82 ženskega in 36 moškega spola, osebe pripadajo različnim ravnim skupinam. Osebe izražajo eno izmed sedmih čustev: jeza, žalost, veselje, strah, gnus, presenečenje in prezir. Posnetek je predstavljen kot zaporedje sličic in v njem nastopa zgolj ena oseba. Na začetku posnetka vsaka oseba izraža nevtralen izraz, ki nato postopoma preide v določeno čustvo, intenziteta izražanja čustva je največja na koncu posnetka. Osebe v bazi so vedno posnete zgolj frontalno, v kontroliranem okolju in v posnetkih tudi ne izvajajo velikih premikov glave. Izrazi niso spontani temveč namensko pozirani, tako da se baza ne smatra kot zahtevna. Dobra lastnost baze je, da vsebuje veliko število oseb, slaba lastnost baze pa je, da testna in učna množica nista določeni, večina člankov tudi ne navede točno kateri posnetki in koliko posnetkov je bilo uporabljenih v učni/testni množici. Zaradi tega je težko primerjati natančnost različnih metod ovrednotenih na bazi med seboj, kot tudi primerjati rezultate lastnih implementacij z rezultati v člankih. Bazo smo uporabili tako, da smo za vsak posnetek izbrali tri sličice iz konca posnetka in nato te skupine sestavljene iz treh sličic naključno razdelili v učno in testno množico. V učni množici smo uporabili 876 sličic, v testni



Slika 4.1: Primeri posnetkov iz baze Cohn Kanade CK+ [20].

množici pa 84 sličic. Prikaz nekaterih posnetkov iz baze je viden na sliki 4.1.

4.2 Opis baze GEMEP-FERA

GEMEP-FERA [27] je del baze GEMEP, ki se je uporabil na tekmovanju FERA-2011 (Facial Expression Recognition and Analysis challenge 2011) [27]. Baza GEMEP vsebuje posnetke 18 različnih čustev, katere so prispevali poklicni igralci, zato se posnetki smatrajo za spontane. Čustva v posnetkih so izražena z različno intenziteto med posnetki, osebe v posnetkih se tudi veliko več premikajo, najbolj izrazito je premikanje glave. Zato se baza smatra za zahtevnejšo. V posnetkih vedno nastopa zgolj ena oseba, posnetki so snemani zmeraj v kontroliranem okolju. Podmnožica baze GEMEP, GEMEP-FERA, vsebuje 155 posnetkov v učni množici in 134 v testni množici, v posnetkih nastopa skupno sedem različnih igralcev, obeh spolov in različnih starosti. Ker imamo točno določeno učno in testno množico, lahko pri tej bazi lažje delamo primerjave lastnih rezultatov z rezultati, ki so navedeni v člankih. Del igralcev v testni množici se ne pojavi v učni množici, tako lahko primerjamo kako se algoritem obnese na obrazih ljudi, na katerih ni bil naučen - nevideni del množice. V posnetkih je prisotnih pet različnih čustev: strah, jeza, veselje, žalost in olajšanje. Nevtralni prikaz ni prisoten v posnetkih, tako da se



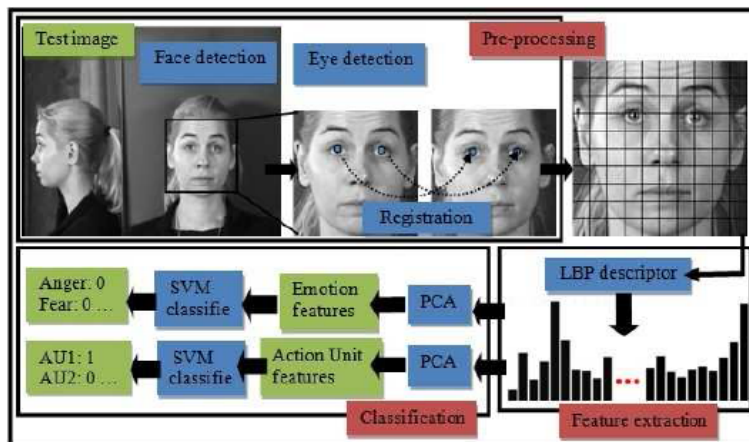
Slika 4.2: Posnetek iz baze GEMEP-FERA [27] in v zgornjem desnem kotu končna slika obraza iz katere beremo značilke.

za vsako sličico posnetka smatra, da predstavlja čustvo. Na sliki 4.2 vidimo primer iz baze GEMEP-FERA.

4.3 Izhodiščna metoda

Organizatorji tekmovanja FERA-2011 so pripravili izhodiščno metodo, s katero so ovrednotili podatkovno množico tekmovanja [27]. Izhodiščna metoda je bila tudi prva metoda, ki smo jo preizkusili. Metoda je obsegala: detekcijo obraza s Haarovi značilkami, nato detekcijo lokacije oči na obrazu s Haarovi značilkami, poravnano obraza glede na zaznano lokacijo centra oči, LBP za izločanje značilk, PCA za zmanjšanje dimenzionalnosti, pri čemer smo ohranili 95% variance in SVM z uporabo RBF jedra in klasifikacijo po strategiji "eden proti ostalim". Na sliki 4.3 je prikaz celotnega poteka izhodiščne metode.

Poravnavo na zaznane oči smo sicer izvedli malce drugače kot avtorji izhodiščne metode. Mi smo z uporabo ene same geometrijske transformacije izvedli: rotacijo s katero sta lokaciji oči na isti višini, skaliranje s katerim



Slika 4.3: Prikaz poteka izhodiščne metode [27].

je razdalja med očmi konstantna in translacijo s katero postavimo končne lokacije obeh oči na isto mesto. Avtorji izhodiščne metode so lokacije oči uporabili zgolj za rotacijo, pri translaciji so zgolj premaknili eno izmed oces na isto mesto. Skaliranja niso izvedli, saj so ocenili da je uporabljena detekcija oči premalo natančna. V primeru, da na sliki obraza nista bili zaznani obe očesi, slike niso vključili v učno množico, prav tako so pri učni množici ročno preverili in odstranili narobe poravnane slike. V našem primeru se je detekcija oči pokazala kot dovolj natančna, res pa je, da smo pri detekciji oči uporabili priporočila iz [10], za največjo možno natančnost. Pri klasifikaciji s SVM smo uporabljali OpenCV-jevo metodo *auto_train* za iskanje naprimernejših parametrov s prečnim preverjanjem.

4.3.1 Rezultati za izhodiščno metodo na bazi Cohn Kanade CK+

Uporabljeno množico smo sestavili tako, da smo iz vsakega posnetka izbrali tri slike, izbirali smo slike na koncu posnetka, z največjo intenziteto. V nekaterih posnetkih na teh končnih slikah niso bile pravilno zaznane oči in smo zato izbrali slike z zaznanimi očmi, ki nastopajo prej v posnetku in se na

njih še zmeraj izraža čistvo. V primeru, da za nek posnetek ni bilo dovolj slik, ga nismo vključili v množico. Težave pri detekciji oči so nastopile predvsem v posnetkih, kadar imajo osebe ob prikazu čustev trdno zaprte oči. Te težave se sicer ne poznajo na rezultatih, saj smo množico tako definirali, da smo uporabili zgolj slike, na katerih sta bili obe očesi uspešno zaznani. Že tukaj je bilo opaziti, da bi za detekcijo oči bila potrebna naprednejša metoda. Pri delitvi v učno in testno množico smo poskrbeli, da so bile tri slike iz vsakega posnetka prisotne zmeraj ali samo v učni ali samo v testni množici. Metoda je dosegla zelo dobre rezultate, 96.42% natančnost oziroma 81 pravilno napovedanih slik, pri 3 nepravilno napovedanih slikah ni bilo zaznano nobeno čustvo. Ker smo že z izhodiščno metodo dobili dobre rezultate, na podatkovni bazi Cohn Kanade CK+ nismo preizkušali nadaljnjih metod.

4.3.2 Rezultati za izhodiščno metodo na bazi GEMEP-FERA

Detekcija obraza in oči je bila na bazi v večini primerov dovolj natančna. Le v štirih primerih je bilo kot obraz z dvema očesa zaznano neko drugo območje, natančneje je bilo to območje vratu, kadar je oseba v posnetku imela glavo obrnjeno navzgor. Dobra lastnost detekcije obraznih značilnk na že prej zaznanem obrazu je, da s tem odstranimo narobe zaznane obraze, saj je verjetnost da bi vsi trije uporabljeni detektorji dali napačno napoved zelo majhna [10]. V enem izmed posnetkov v učni in v treh posnetkih v testni množici nismo dobili končnih slik obraza. Te tri posnetke iz testne množice smo šteli kot napačno napovedane. V teh posnetkih oči niso bile zaznane, saj so igralci gledali vstran, imeli zaprte oči ali so si z rokami zakrivali obraz.

Metoda je dala kar precej boljše rezultate od izhodiščne metode organizatorjev tekmovanja, 79.85% natančnost proti 56%. Rezultat 79.85% je tudi boljši od skoraj vseh ostalih metod, ki so nastopale na tekmovanju razen ene, z eno si deli isti rezultat [27]. Kot možna razloga za neprimerno boljši rezultat sta boljša izbira parametrov za SVM, zaradi uporaba metode *auto_train* ali pa boljša poravnava obraza, predvsem zaradi bolj natančno

zaznane lokacije oči. Namesto uporabljene implementacije LBP v C++, smo značilke pridobili tudi z LBP kodo avtorjev algoritma v Matlabu z enako nadpovprečnim rezultatom, tako da lahko izločanje značilk izključimo kot razlog za razliko v rezultatu. Končen rezultat naše izhodiščne metode na bazi je: 79.85% natančnost, od tega 84.81% natančnostna na videnemu in 72.72% natančnost na nevidnem delu baze. Rezultat izhodiščne metode organizatorjev tekmovanja je: 56% natančnost, od tega 73% natančnost na videnemu in 44% natančnost na nevidnem delu baze. Konfuzijska matrika za izhodiščno metodo je prikazana v tabeli 4.1.

Tabela 4.1: Konfuzijska matrika za izhodiščno metodo (vrstice so dejanske vrednosti, stolpci so napovedi. Torej v tem primeru je bila jeza 25 krat pravilno zaznana, 2 posnetka jeze sta bila nepravilno zaznana kot strah, 5 posnetkov strahu je bilo nepravilno napovedanih kot jeza)

-	Jeza	Strah	Veselje	Olajšanje	Žalost
Jeza	25	2	0	0	0
Strah	5	17	3	0	0
Veselje	0	0	31	0	0
Olajšanje	1	1	1	21	1
Žalost	7	0	0	3	13

4.4 Uporaba EAI predstavitve obraza

Zaradi nadpovprečno dobrega rezultata izhodiščne metode, je bila edina druga metoda, ki smo jo preizkusili, zmagovalna metoda, ki temelji na EAI predstavitvi posnetka [14]. Metoda je obsegala: Detekcijo obraza s Haarovimi značilkami, generiranje EAI predstavitve iz posnetkov za poravnano obraza, LBP ali LPQ za izločanje značilk ter SVM za klasifikacijo z uporabo linearnega jedra in po strategiji "vsak z vsakim". Prednost metode je nadpovprečen rezultat pri nevidnem delu testne množice, slabost pa je, da zaradi počasnosti računanja Sift toka, metoda ni primerna za rabo v realnem

času. Pri metodi smo preizkušali EAI predstavitve na 0, 1 in 2 nivoju, 2 nivo je bil uporabljen za zmagovalni rezultat metode.

4.4.1 Rezultati za metodo z EAI predstavitvijo obraza na bazi GEMEP-FERA

Dobre rezultate smo dobili zgolj z 0 nivojem, saj nam ni uspelo izbrati pravih parametrov pri računanju Sift toka. Primerjava naših EAI predstavitev z EAI predstavitvami avtorjev metode je vidna na sliki 4.4. Pri uporabi EAI predstavitve smo za 0 nivo tudi dobili boljše rezultate z LBP, medtem ko so avtorji metode dobili boljše rezultate z LPQ. Razloga zakaj da LBP boljši rezultat nismo odkrili, poleg C++ implementacije LBP smo preverili tudi rezultat za LBP značilke pridobljene iz kode avtorjev algoritma v Matlabu. Zanimivo je, da če uporabimo SVM s strategijo "eden proti ostalim", dobimo veliko slabše rezultate pri obeh vrstah značilk. Strategija "vsak z vsakim" namreč zmeraj napove vsaj eno čustvo, v tem primeru je to bolje, saj ni posnetkov, ki ne bi pripadali nobenemu razredu, prav tako je pri rabi "vsak z vsakim" manjša možnost, da dobimo več napovedi za en posnetek. LBP značilke dajo pri uporabi EAI predstavitve tudi nadpovprečen rezultat za neviden del množice. Pri napačno klasificiranih primerih pri LBP je pet takih, kjer je SVM glasovanje prineslo enako število napovedi za dva ali več primerov. V takih primerih bi lahko kot končno napoved uporabili rezultat napovedi med primeri z enakim številom glasov. Pri testiranju smo vedno, kadar je bilo več čustev napovedanih za isti primer, tega smatrali kot napačno napovedanega.

Končen rezultat metode na bazi je: za LBP 79.1% natančnost, od tega 78.48% natančnost na videnemu in 80% natančnost na nevidenemu delu množice, konfuzijska matrika za LBP je prikazana v tabeli 4.2. Za LPQ 74.62% natančnost, od tega 78.48% natančnost na videnemu in 69.1% natančnost na nevidenemu delu množice, konfuzijska matrika za LPQ je prikazana v tabeli 4.3. Avtorji metode so navedli rezultat za LBP in LPQ predstavitve na 2 nivoju in sicer LBP 77.38% na celi množici, 87.47% na videnem in

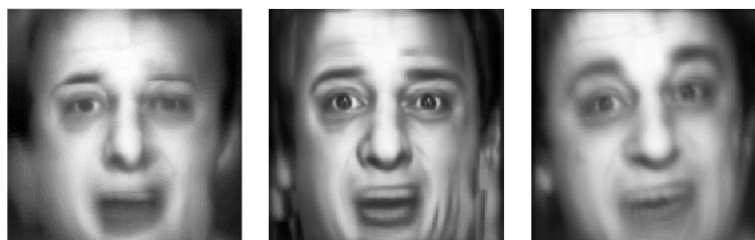
70.64% na nevidenem delu, LPQ 83.78% na celi množici, 96.18% na videnem in 75.23% na nevidenem delu. Za nivo 0 so sicer navedli povprečne rezultate pri 10 kratnem prečnem preverjanju na učni množici in sicer 68% pri LBP in 74% pri LPQ, za primerjavo povprečni rezultat prečnega preverjanja za drugi nivo so bili 84% za LBP in 90% za LPQ .

Tabela 4.2: Konfuzijska matrika za LBP značilke pri rabi EAI predstavitve (vrstice so dejanske vrednost, stolpci so napovedi), primeri z enakim številom napovedi za več različnih čustev niso šteti v tabeli.

-	Jeza	Strah	Veselje	Olajšanje	Žalost
Jeza	25	1	0	0	0
Strah	6	13	4	0	0
Veselje	0	0	28	1	0
Olajšanje	0	0	4	21	1
Žalost	4	0	0	2	19

Tabela 4.3: Konfuzijska matrika za LPQ značilke pri rabi EAI predstavitve (vrstice so dejanske vrednost, stolpci so napovedi), primeri z enakim številom napovedi za več različnih čustev niso šteti v tabeli.

-	Jeza	Strah	Veselje	Olajšanje	Žalost
Jeza	25	1	0	0	1
Strah	3	14	8	0	0
Veselje	0	0	30	1	0
Olajšanje	3	4	3	14	2
Žalost	6	1	0	1	17



Slika 4.4: Primerjava EAI predstavitev na 1 nivoju za isti posnetek. Leva slika je predstavitev avtorjev metode [14], ostali dve sta naši. Sredinska predstavitev ima območja zelo grobih prehodov med slikovnimi elementi, desna predstavitev se je le malenkostno prilagodila glede na sliko avatarja in je zelo podobna povprečni sliki posnetka.

4.5 Preizkus učne množice obeh baz na posnetkih v realnem času

Zato da smo lahko ocenili, ali bo metoda primerna za delo s posnetki okušanja, smo najprej preizkusil izhodiščno metodo na posnetkih iz spletne kamere prenosnika v realnem času. Opravili smo ločena preizkusa za obe bazi. Uporabili smo isti učni množici kot prej, torej k učnima množicama nismo dodajali posnetkov, ki so bili prej v testni množici. Pri preizkusu prek spletne kamere se je detekcija oči s Haarovimi značilkami pokazala kot popolnoma neprimerna. Medtem ko je zaznava obraza delovala dokaj solidno, je bila metoda za detekcijo oči zelo nenatančna. V primeru, da je oseba nosila očala, oči skorajda nikoli niso bile zaznane. Celoten postopek detekcije čustev je bil tudi prepočasen za rabo v realnem času. Zaradi tega smo uporabili CLM za detekcijo obraznih značilk. CLM je veliko bolj natančen in robusten na pogoje v okolju, tudi kadar je oseba nosila očala. Ker CLM sledi obraznim značilkam in ne vsakič preišče slike za lokacijo obraza, je celoten postopek detekcije čustev veliko hitrejši in postane tudi dovolj hiter za delo v realnem času. Za postopek poravnave slike obraze smo potrebovali lokacijo središča obeh oči. CLM sicer ne zazna središča oči, zazna pa lokacije obraznih značilk

okoli robov oči. Ker imajo v implementaciji CLM iste obrazne značilke zmeraj isti indeks, lahko potem glede na njihovo lokacijo določimo središče oces. Tako smo središče desnega očesa določili kot središče obraznih značilk z indeksi: 36, 37, 38, 39, 40 in 41, središče levega očesa pa kot središče obraznih značilk z indeksi: 42, 43, 44, 45, 46 in 47.

Pri preizkusu se je pokazalo, da je metoda naučena na GEMEP-FERA množici zmožna izmed vseh čustev zaznati samo veselje. Kot možna razloga za slabo delovanje sta, da učna množica vsebuje samo pet oseb in da je čustva, ki jih igralci izražajo v posnetkih, težko posnemati. Metoda naučena na Cohn Kanade CK+ je sposobna zaznati vsa čustva na katerih je bila naučena. Nekatera čustva metoda dosti lažje zazna, druga težje in slabše, za nekatera čustva so tudi pogosti konflikti z drugimi čustvi, to pomeni, da je istočasno zaznanih več čustev. Metoda najbolje zazna čustva veselje, presenečenje in gnus. Strah deluje malce slabše, izmed čustev je najtežje zaznati žalost, še posebej kadar oseba nosi očala. Prezir deluje slabše in je pogosto v konfliktu z veseljem in jezo, prezir se namreč zazna, ko ima oseba eno ali obe strani ustnic dvignjeno navzgor. Res pa je, da ima prezir najmanjšo prisotnost v učni množici od vseh čustev. Čustvo jeza je sicer dobro zaznано kadar je v resnici prisotno, vendar je bolj pogosto kot ostala čustva, prisotno v konfliktih, najpogosteje z gnusom in žalostjo, res pa je da je v učni množici prikaz teh čustev dokaj podoben. Za jezo so tudi pogoste lažne napovedi, ko na obrazu ni prisotno nobeno čustvo.

Kasneje pri snemanju posnetkov okušanja hrane in pijače, smo delovanje metode z CLM naučene na Cohn Kanade CK+ uporabljali za ugotavljanje primernosti pogojev za snemanje. Na sliki 4.5 lahko vidimo pravilne napovedi metode.

V tem poglavju smo opisali rezultate različnih metod na bazah Cohn Kanade CK+ in GEMEP-FERA. Metodo, ki se je na bazah najbolj obnesla, smo nato uporabili na posnetkih prek spletne kamere zato, da bi ocenili ali bi bila primerna za rabo na posnetkih, katere bi sami posneli. Ker se metoda ni najbolj obnesla, smo jo zato ustrezno izboljšali in tako dobili končno

metodo. To metodo smo potem uporabili kot osnovo v naslednjem poglavju. Naslednje poglavje opisuje pripravo baze posnetkov odzivov na okus zaužite hrane in rezultate preizkusov na tej bazi.



Slika 4.5: Pravilno zaznana čustva. Zgornja vrsta od leve proti desni: strah, začudenje, veselje. Spodnja vrsta od leve proti desni: jeza, gnus, žalost.

Poglavje 5

Ugotavljanje odziva na okušanje hrane in pijače na lastni bazi

5.1 Priprava in opis baze s posnetki okušanja

Na spletu ni na voljo nobene baze s posnetki okušanja hrane in pijače. Kontaktirali smo tudi avtorje člankov omenjenih v poglavju 1.2 [35] [37] [33], vendar nismo dobili odgovora ali pa avtorji niso imeli soglasja za širjenje posnetkov. Tako smo bazo pripravili sami, po zgledu opisa priprave posnetkov v prej omenjenih člankih. Baza vsebuje 135 posnetkov s petimi različnimi osebami, prikazane so na sliki 5.1. V posnetku zmeraj nastopa samo ena oseba. Tri osebe so moškega spola, dve ženskega, starost oseb v posnetkih se giblje od 28 do 71 let. Na vsako osebo je od najmanj 22, do največ 30 posnetkov. Snemali smo tako prizore okušanja hrane, kot okušanja pijače, v nalogi smo ločeno testirali posnetke okušanja hrane in okušanja pijače.

Posnetke smo snemali v dobri enakomerni osvetlitvi, tako da je obraz okuševalcev frontalno dobro viden. Obraz okuševalca je v posnetku postavljen čimbolj na sredino. Ozadje v posnetku je izbrano tako, da je čimbolj pusto v smislu, da je monotone barve in nima nobenih predmetov v ozadju,



Slika 5.1: Osebe, ki nastopajo v naši podatkovni bazi.

ki bi lahko bili napačno zaznani kot obraz. Posnetke smo snemali z ročno kamero z uporabo stojala, v ločljivosti 1440×1080 . Kamera je posnetke shranila v formatu s končnico .MTS - Blu-ray Disc Audio-Video (BD-AV) MPEG-2 Transport Stream. Posnetek se začne pred okušanjem hrane oziroma pijače. Ob koncu okušanja, so okuševalci izrekli besedo stop, s čimer se je posnetek končal. Pri okušanju pijače so, po zgledu [36], okuševalci pri pitju uporabili slamico zato, da je bilo čim manj prekrivanja obraza pri pitju. Okuševalci so spili določeno količino pijače, umaknili slamico in pustili pijačo par sekund v ustih zato, da je bil na obrazu viden odziv na okus. Obraz so med snemanjem imeli frontalno obrnjen proti kameri. Pazili smo, da se okuševalci pri vnosu pijače niso sklanjali naprej, temveč so si pijačo prinesli z roko bližje k obrazu in da so po zaužitju čimprej odmaknili slamico iz obraznega dela. Sicer smo se pri snemanju trudili, da so bili izrazi čimbolj spontani. Pri zaužitju hrane so okuševalci vnesli hrano večinoma z rokami. Tudi tukaj smo pazili, da je bilo pri zaužitju, prekrivanje obraza čim krajše. Pazili smo tudi, da so bili obrazi snemani frontalno in da se okuševalci pri zajemu hrane niso sklanjali naprej. Pri okušanju hrane smo okuševalcem pustili, da so hrano prežvečili poljubno dolgo. Posledično so bili posnetki okušanja hrane daljši in s tem množica slik pri testiranju večja. Zato, da posnetki okušanje hrane niso bili predolgi, smo uporabili manjše kose hrane.

Po zaužitju hrane in pijače so morali okuševalci podati oceno. Po zgledu člankov iz poglavja 1.2 smo uporabili lestvico od 1 do 7. Ocena 1 pomeni zelo neprijeten okus, 7 pa zelo prijeten okus. Oceno 4 bi uporabili kadar je okus pri zaužitju nevtralen, na primer pri pitju vode. Kasneje v nalogi smo posnetke z ocenami od 1 do 3 smatrali kot negativne odzive, posnetke z ocenami od 4 do 7 pa kot pozitivne odzive. Pri snemanju smo za vsako osebo posneli negativne in pozitivne odzive, tako za pitje pijače, kot tudi za okušanje hrane. V večini primerov smo pri posamezni osebi posneli dva posnetka z določeno vrsto hrane oziroma pijače. Med posnetki z različno hrano in različnimi pijačami, smo okuševalcem dali po želji kratek premor, še posebej po okušanju zelo neprijetnih živil. Med okušnji so si okuševalci

pogosto tudi sprali usta z vodo. Za razliko od nekaterih člankov iz poglavja 1.2 [36] [37] [33] se nismo usmerili na točno določeno vrsto hrane in pijače. Pri pijači smo uporabili vodo, mleko, jogurt, sok stisnjen iz limone, sadne sokove, zelišče nim v prahu raztopljeno v vodi (to je sprožilo pri okuševalcih od vseh pijač/hran najbolj negativne reakcije), kot tudi nekatere alkoholne pijače recimo pivo in liker. Pri hrani smo uporabili koščke raznovrstnega sadja, koščke mesa, sir, koščke čokolad in piškotov. Izbira je bila odvisna predvsem od preferenc okuševalca v času snemanja.

5.2 Prvotni rezultati na bazi

5.2.1 Pretvorba posnetkov v zaporedje slik obrazov

Najprej smo z uporabo CLM in poravnave glede na zaznane oči pretvorili posnetke v seznam zaporednih slik obrazov. Uporabili smo slike iz vseh 135 posnetkov. Kot v prejšnjemu poglavju smo sliko obraza pomanjšali na velikost 200×200 . Ročno smo pregledali nastala zaporedja slik. Pri večini posnetkov je bilo končno zaporedje slik dobro, v nekaterih primerih pa so se pojavile napake.

Napake so se pojavile predvsem zaradi zamika obraznih značilk pri prekrivanju obraza torej, ko so okuševalci vzeli hrano. V nekaterih primerih so se kljub zamiku lokacije obraznih značilk pri prekrivanju, te poravnale nazaj na pravo mesto. Zgodilo se je tudi, da se vse obrazne značilke, ki jim CLM sledi, niso poravnale nazaj na pravo mesto, predvsem značilke na dnu roba obraza in okoli ust so ostale na previsokem mestu, vendar to ni imelo vpliva pri uporabljeni poravnavi, ki je odvisna zgolj od lokacije oči. V primeru napak pri določenih posnetkih smo ponovno pognali CLM in poravnavo na teh posnetkih. Uporabljena implementacija CLM omogoča, da lahko med potekom sledenja, lokacijo obraznih značilk ponovno incializiramo, kar smo tudi uporabili v nekaterih posnetkih, kadar so se pojavile napake zaradi prekrivanja.

Poravnava na zaznane oči je dala v večini primerov dobre končne slike

obraza, sličice obraza s prekrivanjem zaradi roke so pogosto bile zamaknjene. Na končnem zaporedju slik obraza je tudi opazen pojav približevanja in oddaljevanja od obraza, to lahko vidimo na sliki 5.2. Razlog, da to tega pride, je v zaznani lokaciji centra oči. Ko ima oseba zaprte oči, recimo zaradi mežikanja, sta zaznani levi in desni center oči bližje drug drugemu. Zato pri zapiranju oči izgleda, da se rahlo približujemo obrazu, pri odpiranju oči pa, da se rahlo oddaljujemo od obraza. CLM in poravnava na oči sta na bazi v večini primerov dobro poravnali obraz, vendar bi bilo vseeno smiselno uporabiti bolj robustno metodo detekcije obraznih značilk. V primeru bolj robustne detekcije obraznih značilk, bi lahko uporabili tudi bolj napredno metodo poravnave obraza, ki uporablja lokacije večih obraznih značilk.

Iz dobljenih seznamov sličic posnetkov, smo potem za vsak posnetek določili mesta, od katere do katere sličice bomo uporabili posnetke v prečnem preverjanju. Z ročnim pregledom smo začetek določili kot prvo mesto, ko ni bilo več prekrivanja, oziroma v primerih, ko je prekrivanje povzročilo slabo poravnavo na končni sliki obraza, kot prvo sličico s pravilno poravnavo, v teh primerih smo izpustili nekaj sličic. Konec smo določili kot mesto par sličic preden je oseba izgovorila besedo stop, torej potem, ko je pogoltnila hrano in pijačo.

5.2.2 Rezultati prečnega preverjanja

Vsak posnetek in s tem vse pripadajoče sličice je imel oznako negativnega odziva ali pozitivnega odziva, torej smo imeli dvorazredni klasifikacijski problem. Uporabili smo LBP za izločanje značilk in en SVM z RBF jedrom za klasifikacijo. Posnetke smo ločili v dve množici glede na to ali je šlo za posnetke okušanja hrane ali pa posnetke s pitjem pijač. Množici imata približno enako število posnetkov, 66 posnetkov pri pijači in 69 posnetkov pri hrani. Kljub temu ima množica okušanja hrane neprimerno več sličic 17861, medtem ko ima množica pijač 6206 sličic, torej skoraj trikrat manj, razlog za to je v daljših posnetkih pri okušanju hrane. Obe množici smo večkrat ovrednotili s trikratnim prečnim preverjanjem.



Slika 5.2: Prikaz približevanja in oddaljevanja od obraza, zaradi zapiranja in odpiranja oči, na osmih zaporednih končnih sličicah obraza.

Posnetke smo naključno razdelili v tri množice, vse sličice nekega posnetka so se lahko pojavile zgolj v eni izmed treh množic. Tri dobljene množice se sicer lahko precej razlikujejo v številu sličic. Eden razlog za to je v različnem številu sličic po posnetkih. Nadalje smo pri delitvi posnetkov upoštevali, da ima vsaka izmed desetih kombinacij osebe in vrste odziva (pozitivni in negativni) čimbolj enakomerno razporeditev. Ker delimo posnetke v tri množice to pomeni, da ima neka množica za vsako poljubno kombinacijo lahko največ en posnetek več ali manj, kot ostali dve množici. Ker smo kombinacije delili zmeraj po istem vrstnem redu, torej najprej prvi množici, potem drugi in potem tretji, to pomeni da ima druga množica zmeraj manj ali enako število posnetkov kot prva množica, prav tako ima tretja množica zmeraj manj ali enako število posnetkov kot druga množica.

V treh iteracijah prečnega preverjanja smo vsako izmed treh množic izbrali enkrat kot učno, ostali dve pa smo združili v testno množico, torej obratno, kot je navada pri prečnem preverjanju. Tako je vsak posnetek nastopil dvakrat v testni množici. Razlog za to je bilo hitrejše učenje, saj učenje SVM z naraščanjem števila učnih primerov deluje počasneje, SVM smo na-

mreč preizkušali z več različnimi parametri. Na koncu smo kot optimalna parametra izbrali 2.5 za C parameter in 0.50625 za gamma parameter. V testni fazi smo za vsak posnetek posebej šteli število pozitivnih in negativnih napovedi po sličicah posnetka. Kot napovedani odziv posnetka smo izbrali tistega z večjim številom napovedi po sličicah. Če se je napovedan odziv ujema z odzivom posnetka, smo posnetek šteli kot pravilno napovedan, sicer smo ga šteli kot nepravilno napovedanega. V primeru, da bi za nek posnetek dobili enako število pozitivnih in negativnih napovedi po sličicah, bi ta posnetek šteli kot nepravilno napovedan. Zaradi tega kako določimo končno napoved se lahko zgodi, da je razlika med številom pozitivnih in negativnih napovedi po sličicah pri nekemu posnetku zelo podobna in je potem posnetek štet kot pravilen ali nepravilen zgolj zaradi minimalne razlike v napovedih.

Rezultat prečnih preverjanj je zelo dober pri pijači in dokaj soliden pri hrani. Navedeni rezultati veljajo za 10 ponovitev prečnih preverjanj z izbranimi optimalnimi parametri za SVM in so zaokroženi na celo število. Pri pijači je povprečna natančnost 10 prečnih preverjanj 92%, najnižja dosežena natančnost je bila 87%, najvišja pa 94%. Pri hrani je povprečna natančnost 10 prečnih preverjanj 81%, najnižja dosežena natančnost je bila 76% najvišja pa 84%. V prečnih preverjanjih je pri hrani večje število False Positive napak (posnetki z negativnim odzivom so narobe klasificirani kot pozitivni odzivi), pri pijači pa je ravno obratno, tam prevladujejo False Negative napake (pozitivni odzivi so narobe klasificirani kot negativni). Ena izmed možnih razlag za to razliko je, da je pri posnetkih s hrano razmerje med posnetki z negativnim odzivom in pozitivnim odzivom ni enakomerno, pozitivnih je več in sicer 41 proti 28 negativnim, medtem ko je pri pijači število obeh odzivov 33. V nobenem izmed prečnih preverjanj sicer ni bilo prevelikih razlik med razmerjem pri številu False Positive in False Negative napak.

5.2.3 Analiza napak pri prečnem preverjanju

Poleg rezultatov smo tudi podrobno analizirali napake prečnih preverjanj, iskali smo vzorce pri primerih, ki so bili najpogosteje narobe napovedani in

primerih, ki so zmeraj prav napovedani. Pri hrani smo ugotovili, da več kot polovico napak, 140 izmed 261 napak, sestavlja množica zgolj 11 posnetkov. Vseh teh 11 posnetkov je bilo vsaj 10 krat narobe napovedanih, torej vsaj polovico časa. Pri posnetkih ne prevladuje nobena oseba, 8 jih prikazuje negativen odziv na okus hrane. Če gledamo posnetke z vsaj 11 napakami, teh je 8, ugotovimo da vsi razen enega prikazujejo negativen odziv na okus hrane. To potrjuje opažanje, da pri hrani prevladujejo False Positive napake nad False Negative napakami. Za 8 izmed 11 najpogostejše narobe napovedanih posnetkov, so okuševalci oceno dali 2 ali 3. Pri ogledu teh 8 posnetkov smo ugotovili da je za 6 izmed njih značilno, da je na obrazu okuševalca napram pravilno napovedanimi posnetki negativnega odziva, težko ugotoviti negativen odziv oziroma se na obrazu ne kaže negativen odziv. Primerjava posnetka negativnega odziva z nizko natančnostjo in negativnega odziva z visoko natančnostjo je vidna na sliki 5.3. Možno je tudi, da je okuševalec podal napačno oceno in ima tako posnetek zabeležen napačen odziv. Pri enem izmed teh posnetkov, smo opazili problem zaradi večkratnih snemanj okušanja iste hrane. Ta posnetek je bil tretji zaporedni posnetek snemanja odziva na isto hrano. Vsi 3 posnetki so bili ocenjeni z negativnim odzivom medtem, ko je pri prvih dveh posnetkih viden izrazit negativen odziv, pri tretjemu posnetku odziv ni več toliko izrazit, okuševalec se je navadil na okus. Tako sta bila prva dva posnetka zmeraj prav napovedana kot negativna odziva, medtem ko je bil tretji 14 krat narobe napovedan. Pri ostalih 2 izmed 8 posnetkov, se kaže neka druga slabost naše metode. Pri obeh posnetkih je sicer negativen odziv jasno viden, vendar se pojavi šele proti koncu okušanja, v začetku okušanja negativen odziv ni opazen. Tako imata ta dva posnetka pogosto pri napovedih po sličicah razmerje med možnima odzivoma okoli 40% proti 60%, na koncu pa se posnetek klasificira v določeni odziv, zgolj zaradi nekaj napovedi po sličicah več.

Pri pijači je 44 posnetkov vedno brez napak, torej točno dve tretjini, medtem ko je bilo pri hrani takih posnetkov slaba tretjina. Pri pijači izstopata po številu napak 2 posnetka, oba sta namreč zmeraj narobe napovedana,



Slika 5.3: Sličice iz zgornje vrste so iz posnetka negativnega odziva s tremi nepravilnimi napovedi, obrazni izraz neprijetnosti okusa je jasno viden. Sličice iz spodnje vrste so iz posnetka negativnega odziva, ki je bil zmeraj napačno napovedan, nikjer v posnetku ni mogoče opaziti obraznega izraza, ki bi kazal na neprijetnost okusa.

ostali posnetki imajo največ 10 nepravilnih napovedi, torej so vsaj polovico časa prav napovedani. Eden izmed posnetkov predstavlja pozitiven odziv in ima oceno 5, drugi pa predstavlja negativni odziv in ima oceno 3. Posnetka pripadata različnim osebam. Oseba z narobe napovedanim posnetkom z negativnim odzivom ima preostale posnetke z negativnim odzivom zgolj z ocenami 1 in 2, vsi ti posnetki prikazujejo dosti bolj opazen negativni odziv. Tukaj je torej možen razlog za napake pomanjkanje posnetkov iste vrste odziva s podobno intenzivnostjo prikaza odziva. Oseba z narobe napovedanim posnetkom s pozitivnim odzivom ima v svoji množici zgolj še 2 druga posnetka s pozitivnim odzivom, oba imata oceno 6. Eden izmed teh posnetkov je imel 9 napačnih napovedi, zgolj 3 posnetki so imeli več napak, drugi pa 4 napačne napovedi, kar je tudi precej glede na število napak pri ostalih primerih. Ker ima ista oseba 7 posnetkov pitja pijača z negativnim odzivom, je tukaj razlog za napačne napovedi najverjetneje v neenakomernem razmerju med številom pozitivnih in številom negativnih odzivov v posnetkih osebe. Če pregledamo ostale posnetke, ki so imeli največje število napak, ugotovimo da prevladujejo posnetki s pozitivnim odzivom, torej to potrjuje opažanje, da pri pijači prevladujejo False Negative napake nad False Positive napakami. Izjema je posnetek s 10 napakami, ki je bil tretji po številu napak. Posnetek je imel oceno 1, vendar pa na obrazu okuševalca ni bilo videti izrazitega negativnega odziva. Ta posnetek je pri poteku snemanja sledil posnetku iste pijače, z istim negativnim odzivom, prejšnji posnetek je imel 3 napake. Kot pri podobnem primeru pri hrani je tudi tukaj možna razlaga, večkratno zaporedno snemanje in s tem zmanjšani odziv na neprijeten okus v kasnejših snemanjih.

Poleg štetja napak smo po posnetkih šteli tudi povprečno razmerje med nepravilno napovedanimi sličicami in vsemi sličicami posnetka. Po pričakovanju je, s številom napak posnetka v prečnemu preverjanju, razmerje naraščalo, pri posnetkih brez napak je bilo razmerje blizu 0. Vseeno pa smo našli posnetek brez napak, ki je imel povprečen odstotek napačnih napovedih po sličicah okoli 46%. Pri temu posnetku je šlo za negativen odziv na pitje

žgane pijače, odziv pa je nastopil šele v zadnji polovici posnetka, po kaki sekundi. Podobno opažanje o poznem odzivu smo opazili že pri hrani, le da je pri pijači večja možnost za napako, saj so posnetki krajši. Zaradi tega bi bilo morda smiselno pri določenih vrstah hrane in pijače, dati različno težo napovedim glede na čas, ali pa pri delitvi posnetka izpustiti nekaj začetnih sličic. Povzetek rezultatov na bazi in sestavi baze, glede na pijačo in hrano je prikazan v tabeli 5.1.

Tabela 5.1: Primerjava velikosti množice in rezultatov prečnega preverjanja po pijači in hrani.

-	Pijača	Hrana
Povprečna natančnost	92 %	81 %
Najmanjša natančnost	87 %	76 %
Največja natančnost	94 %	84 %
Št. posnetkov	66	69
Št. sličic	6206	17861
Št. pozitivnih posnetkov	33	41
Št. negativnih posnetkov	33	28

5.3 Nadaljni poizkusi in izboljšave

5.3.1 Zamenjava CLM z algoritmom za detekcijo obraznih značilk iz knjižnice dlib.

CLM je pokazal pri detekciji in sledenju obraznih značilk dve pomanjkljivosti. Prva je približevanje in oddaljevanje od obraza pri zapiranju oči. Druga pomankljivost, ki je bolj pomembna pa je, da lahko, zaradi nastopa prekrivanja obraza, pride do zamika zaznane lokacije obraznih značilk, katerim sledimo, na napačna mesta. Uporabljena implementacija CLM sicer omogoča, da namesto stalnega sledenja, vsakih nekaj sličic ponovno zaznamo obraz, kar bi omililo to pomankljivost. Vendar smo kot boljšo rešitev uporabili detektor

obraza in obraznih značilk iz knjižnice dlib. Glavna razlika med njim in CLM je, da namesto sledenja obraznih značilkam, vsakič zazna obraz in na njem določi obrazne značilke. Kljub temu, da obraznih značilkam ne sledimo, je algoritem dovolj hiter za delo v realnem času. V nekaterih primerih zaradi prekrivanja ni bil zaznan obraz, tako da je detektor iz dlib, vrnil za določene posnetke krajše zaporedje slik. Zaradi rabe detektorja iz dlib, je tudi pri zapi-ranju oči veliko manj opazno približevanje in oddaljevanje od obraza. Primer tega je viden na sliki 5.4. Model uporabljen z dlib detektorjem sicer zazna 68 obraznih značilk, vendar so indeksi za obrazne značilke okoli obeh oces, isti kot pri modelu pri CLM. Središče desnega očesa se določi kot središče točk z indeksi: 36, 37, 38, 39, 40 in 41. Središče levega očesa se določi kot središče točk z indeksi: 42, 43, 44, 45, 46 in 47.

5.3.2 Uporaba dveh SVM klasifikatorjev namesto enega

Zaradi rabe enega samega SVM klasifikatorja bo vsaka sličica v posnetku označena kot pozitivni ali negativni odziv na okus hrane. Možno pa je, da sličica v posnetku ne predstavlja niti pozitivnega niti negativnega odziva. Recimo, če bi prezkodaj ocenili začetek prekrivanja, bi klasifikacijo vršili tudi na sličicah s prekrivanjem. Ob konca posnetka potem, ko je oseba zaužila hrano, se tudi pojavijo sličice, ko oseba ne izraža odziva na hrano. Zato smo se odločili za preizkus z uporabo dveh SVM klasifikatorjev. Eden izmed klasifikatorjev bi zaznal zgolj prisotnost pozitivnega odziva na hrano, drugi pa le prisotnost negativnega odziva na hrano.

Najprej smo preizkusili delovanje dveh klasifikatorjev, kjer smo za oba klasifikatorja uporabili identične učne množice, kot bi pri rabi enega kla-sifikatorja. En klasifikator smo označili kot prijetnega, torej je pozitivna napoved klasifikatorja pomenila pozitivni odziv osebe na okus hrane. Dru-gega pa kot neprijetnega, torej je pozitivna napoved klasifikatorja pomenila negativen odziv osebe na okus hrane. Končno napoved posnetka smo določili glede na to, kateri klasifikator je imel več pozitivnih napovedi. V primeru enakega števila napovedi, bi posnetek označili kot napačno napovedanega.



Slika 5.4: Primerjava razlik zaradi lokacij zaznanih oči med CLM in dlib, na končnih slikah obraza, na istih zaporednih sličicah istega posnetka. Prva in tretja vrsta CLM, druga in zadnja vrsta dlib.

Istočasno smo med potekom prečnega preverjanja šteli število primerov, ko sta prijetni in neprijetni klasifikator hkrati vrnila negativno napoved in ko sta hkrati vrnila pozitivno napoved. Tako pri hrani, kot pri pijači ni bilo sprememb v končnih rezultatih. Pri obeh je bilo število primerov, kjer sta oba klasifikatorja istočasno imela pozitivno ali negativno napoved majhno, do največ 100 za celotni potek enega trikratnega prečnega preverjanja.

Ker delovanje dveh klasifikatorjev neodvisno drug od drugega, ne spremeni končnega rezultata, smo poizkusili še z dodajanjem dodatnih primerov v njuni učni množici. Naša baza posnetkov ima zgolj označbo končnega odziva na okus, ni pa označb, kateri deli posnetka izražajo ta odziv na okus. V primeru, da bi imeli to označbo, bi dele posnetkov kjer se odziv ne izraža, uporabili v obeh SVM klasifikatorjih kot negativne primere. Ker tega nimamo, smo kot dodatne negativne primere za SVM klasifikatorja vzeli prvih pet sličic vsakega posnetka, ki je v dani iteraciji prečnega preverjanja v učni množici. Tako pri hrani, kot tudi pri pijači, je poraslo število primerov, kot sta oba klasifikatorje imela hkrati isto napoved. Število primerov ko sta oba klasifikatorja imela pozitivno napoved je bilo največ 200, večje povečanje je bilo v številu primerov, ko sta oba klasifikatorja imela negativno napoved, od 500 do 1000 primerov pri pijači in pri hrani okoli 2000 primerov. Pri hrani ni bilo razlik v končni natančnosti prečnega preverjanja ob isti razporeditvi posnetkov. Pri pijači je bila v nekaterih primerih končna natančnost boljša, v nekaterih primerih slabša. Razlike v natančnosti so bile sicer majhne, največ tri pravilno napovedane posnetke razlike.

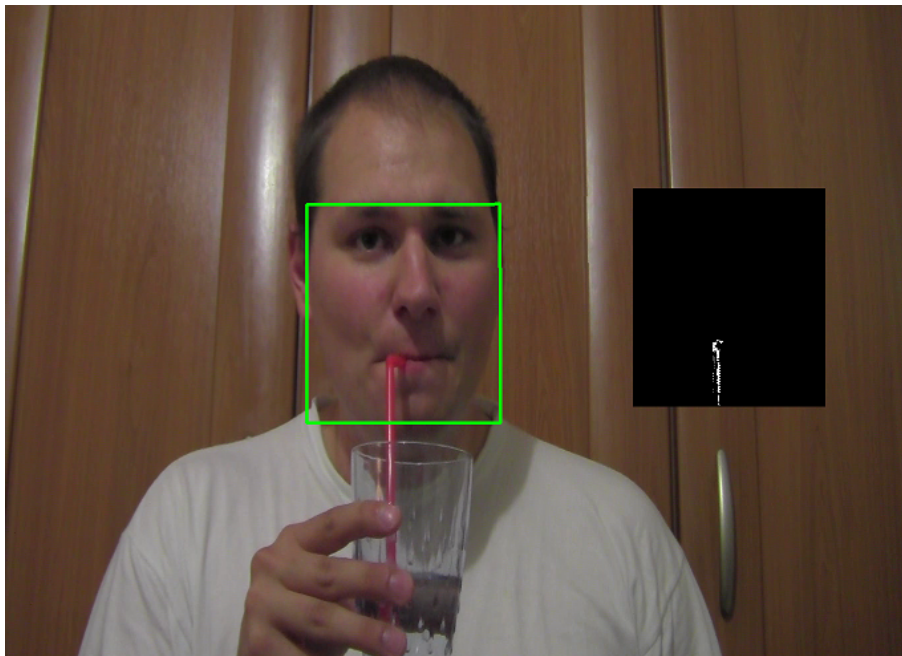
Z uporabo dveh klasifikatorjev sicer nismo dosegli izboljšave, vendar bi bila uporaba dveh klasifikatorjev namesto enega vseeno smiselna, če bi se v posnetkih meja konca prekrivanja določila prezgodaj, če bi v klasifikaciji nastopale tudi sličice potem, ko je oseba pogoltnila hrano, ali pa bi vrsta testirane hrane oziroma pijače sprožila na obrazu odziv šele po par sekundah, torej kasneje kot običajno. Pri dveh klasifikatorjih bi bilo koristno tudi, če bi baza posnetkov imela tudi točno določena območja, kdaj v posnetku se na obrazu izraža odziv na okus hrane.

V tem poglavju smo opisali pripravo baze posnetkov in rezultate prečnega preverjanja na tej bazi. Podrobno smo tudi analizirali napake. Ugotovili smo da lahko metode za prepoznavo obraznega izraza uspešno uporabimo na problemu ugotavljanja pozitivnega ali negativnega odziva na okus hrane. Napake nastanejo predvsem v posnetkih, kjer negativni odziv na okus hrane ni viden na sliki obraza. Sličice posnetkov, ki smo jih uporabili v učni in testni množici, smo ročno določili. Ta postopek bi lahko avtomatizirali, z ugotavljanjem prekrivanja obraza. Naslednje poglavje opisuje različne pristope za ugotavljanje prekrivanja in njihovo natančnost.

Poglavje 6

Iskanje konca prekrivanja obraza pri vnosu hrane in pijače

V prejšnjem poglavju smo pri klasifikaciji uporabili zgolj obrazne sličice po koncu prekrivanja, torej ko je okuševalec zaužil hrano in pijačo. Mejo, kje se prekrivanje konča, smo določili z ročnim pregledom. Sličice pred vnosom hrane je nesmiselno uporabiti, prav tako slike med vnosom hrane zaradi prekrivanja obraza. Zaradi tega se je pojavila potreba po avtomatskem ugotavljanju, kdaj je nastopilo in prenehalo prekrivanje obraza zaradi vnosa hrane oziroma pijače. V nekaterih primerih, kot je na primer prekrivanje zaradi slamice, lončka ali vilice, bi lahko konec prekrivanja določili glede na segmentacijo barve v HSV barvnem prostoru. Predmet bi moral biti v tem primeru take barve, ki ne more nastopati na sliki obraza, tako da je to zelo specifična rešitev, primer lahko vidimo na sliki 6.1. Drugače je prekrivanje obraza z roko težko določiti, saj je barva rok enaka barvi obraza. Zaradi različnih možnih načinov prekrivanja, ki lahko nastanejo pri okušanju, smo poizkušali poiskati metodo, ki bi bila čimbolj splošna. Za metodo ni potrebno, da natančno določi začetek prekrivanja, zgolj da dovolj krat zazna prekrivanje, da vemo kdaj je obraz zakrit zaradi zaužitja. Prav tako ni po-



Slika 6.1: Primer segmentacije slamice glede na rdečo barvo, segmentacijo je bila uporabljena na delu obraza znotraj zelenega pravokotnika. Območje obraza smo določili kot bounding box okoli točk, katere je našel detektor obraznih značilk iz dlib.

trebno, da metoda natančno določi kdaj je prekrivanje končalo. Dovolj je že, da zazna zadnjič prekrivanje, ko se roka odmika proč od obraza. Mejo za konec lahko nastavimo tudi nekaj sličic po tem, ko je metoda zadnjič zaznala prekrivanje. V posnetkih odziv na okus nastopi na obrazu zmeraj nekaj sličic po zaužitju oz. koncu prekrivanja. Težave bi lahko nastopile, če bi metoda velikokrat zaznala prekrivanje kadar to ni prisotno, potem bi lahko začetek posnetka določili prepozno.

6.1 Iskanje z detektorjem obraznih značilk, ki ugotavlja ali so značilke prekrите

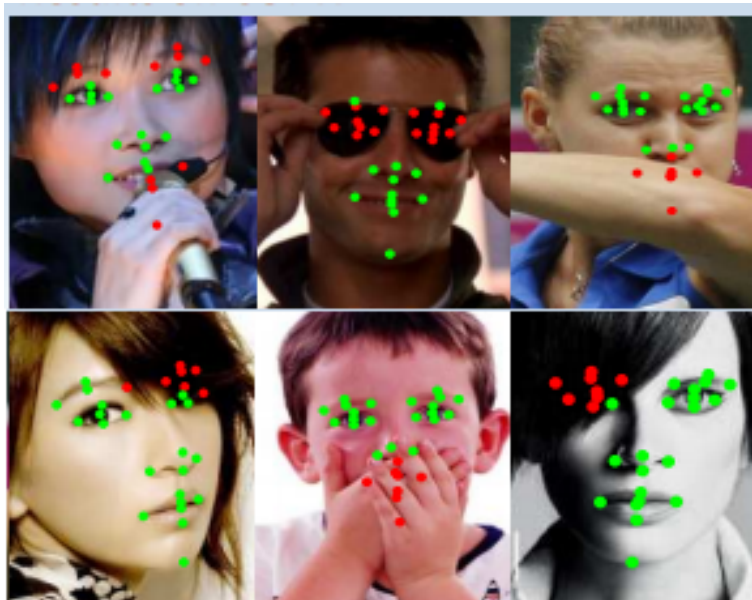
V zadnjemu času se je pojavilo nekaj detektorjem obraznih značilk, ki ugotavljajo ali so značilke prekrите. V našem primeru bi nas zanimalo ali so prekrите obrazne značilke okoli ust in v spodnjem delu obraza. Ti detektorji večinoma razširjajo že od prej prisotne detektorje tako, da ti poleg lokacije obraznih značilk tudi ocenijo ali so te prekrите. Poleg ocene prekrivanja naj bi ti detektorji bolje določili lokacijo značilk ob prekrivanju. V prikazanih slikah so rdeče obarvane obrazne značilke napovedane kot prekrите, zelene obarvane pa niso napovedane kot prekrите.

6.1.1 Robustna ocenitev poze z vrsto regresorjev (RCPR)

Opis metode

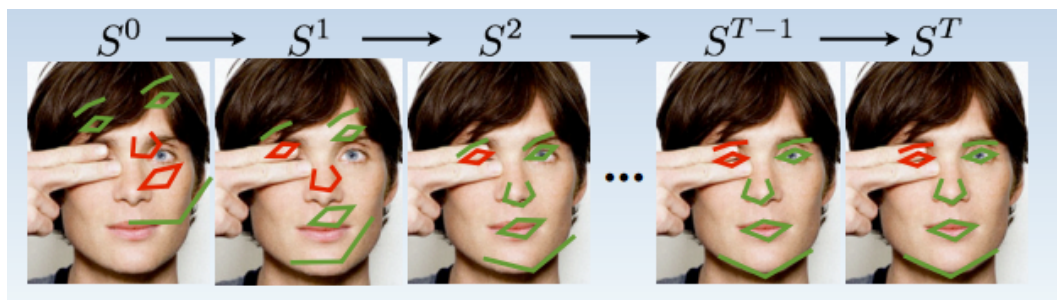
Metoda RCPR (Robust Cascaded Pose Regression) [41] temelji na izboljšavi že obstoječe metode za detekcijo obraznih značilk, ki uporablja vrsto regresorjev, delovanje vrste je prikazano na sliki 6.3. Vrsta deluje tako, da vsak naslednji regresor v vrsti izboljšuje napovedi regresorjev pred njim. Ta metoda se sooča s težavami, kadar pride do velike variacije v lokaciji obraznih značilk ali pa pri prekrivanju obraznih značilk. Metoda RCPR prejšnjo metodo izboljša tako, da upošteva že v času učenja vidnost obrazne značilke, torej imamo za vsako obrazno značilko tri podatke namesto dveh. Vidnost obrazne značilke je definirana kot binarna vrednost, torej 0 ali 1. Prikaz napovedi metode je viden na sliki 6.2.

Med potekom algoritma se sliko obraza razdeli v mrežo velikosti 3×3 . Med iteracijami se za vsako izmed teh območij oceni trenutna povprečna vidljivost območja. Metoda RCPR v vsaki iteraciji vrste, namesto enega regresorja uporabi več regresorjev. Ti regresorji se med sabo razlikujejo glede na uporabljene vizualne značilke, vsak izmed regresorjev lahko vzame le vizualne značilke iz zgolj enega izmed devetih območij mreže obraza. Primer

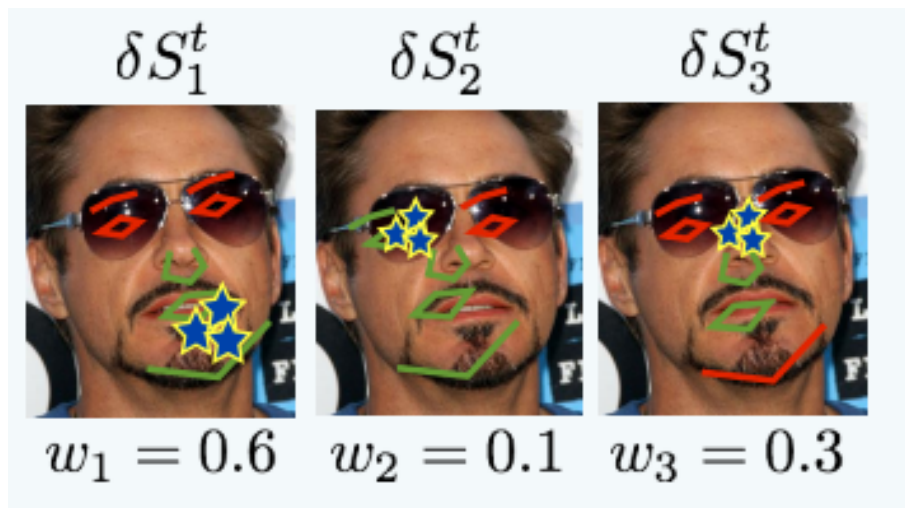


Slika 6.2: Primeri pravih napovedi z metodo RCPR, kot so jih avtorji metode prikazali v članku [41].

tega je viden na sliki 6.4. Ko dobimo za vse regresorje v iteraciji napovedi, povprečimo napovedi glede na ločene uteži za vsak regresor. Uteži so obratno sorazmerne glede na vidljivost območja, iz katerega je regresor izbral vizualne značilke. Ideja algoritma je, da dosegamo podobne napovedi z vizualno različnimi regresorji, hkrati pa pri napovedih damo več zaupanja območjem, ki niso prekrita. Pri določanju lokacije obraznih značilk nastane težava, saj se lahko trenutne razporeditve obraznih značilk precej razlikujejo med obrazi, vendar pa bi želeli, da so izbrane lokacije uporabljenih vizualnih značilk čimbolj podobne med možnimi razporeditvami obraznih značilk. Zato so avtorji metode RCPR določili lokacijo uporabljenih vizualnih značilk, kot točko na črti med dvema trenutnima lokacijama obraznih značilk. To lahko vidimo na sliki 6.5. Na koncu učenja algoritma se določi meja glede na izračunane verjetnosti za vidljivosti, meja pove ali je glede na napoved, obrazna značilka vidna ali pa ne.



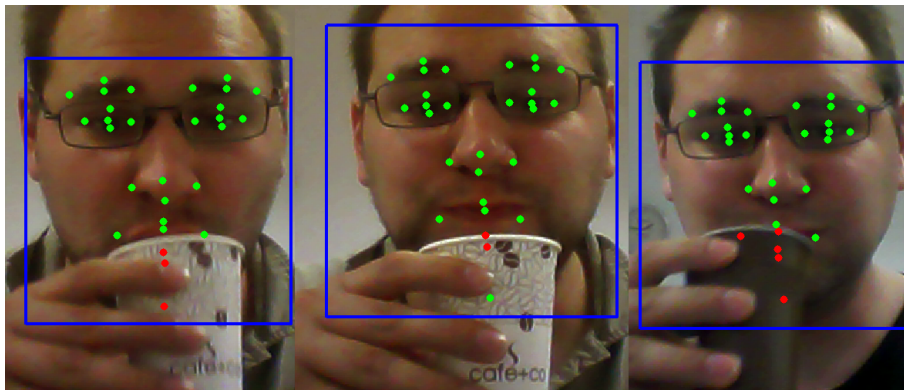
Slika 6.3: Primer delovanja vrste regresorjev pri metodi RCPR [42].



Slika 6.4: Metoda RCPR uporablja več regresorjev v posamezni iteraciji, njihove napovedi upoštevamo z različnimi utežmi. Avtorji so dobili dobre rezultate že z rabo treh regresorjev v posamezni iteraciji vrste [42].



Slika 6.5: Zgornja vrsta: metoda na kateri temelji RCPR določi lokacijo značilk tako, da izračuna transformacijo, ki preslika začetno razporeditev obraznih značilk v trenutno razporeditev obraznih značilk. Z uporabo transformacije se potem lokacije uporabljenih vizualnih značilk, preslikajo iz začetne razporeditve obraznih značilk glede na trenutno razporeditev obraznih značilk. Spodnja vrsta: metoda RCPR določi lokacijo uporabljene vizualne značilke kot točko na črti, ki poteka med trenutno napovedano lokacijo dveh obraznih značilk. Ta način določanja lokacije vizualnih značilk je bolj robusten na spremembe v razporeditvi obraznih značilk in tudi hitrejši [41].



Slika 6.6: Primeri prekrivanja z lončkom kjer je metoda RCPR pravilno napovedala prekrivanje obraznih značilk.

Preizkus metode

Metoda se je na našem preizkusu slabo obnesla. Metodo smo preizkusili na posebej pripravljenih posnetkih iz spletne kamere, ki so vsebovali prekrivanje z lončkom ali z roko in na nekaterih posnetkih okušanja hrane iz naše podatkovne baze. Model, ki je bil vključen zraven izvirne kode metode, zaznava manjše število obraznih značilk 29, okoli ust je značilk šest in ena sama značilka na spodnjem robu obraza, natančneje na dnu brade. Torej je na območju, kjer nas prekrivanje zanima, majhno število obraznih značilk. V nekaterih primerih prekrivanja, pri pitju z lončkom, je metoda sicer prav zaznala prekrivanje, taki primeri so vidni na sliki 6.6. Metoda RCPR skoraj nikoli ni zaznala prekrivanja rok. Razlog za slabo delovanje metode je mogoče tudi v napačni izbiri parametrov.

Kljub slabemu rezultatu, bi bila metoda mogoče primerna za uporabo, če bi jo naučili na učni množici z večjim številom anotiranim obraznih značilk na obrazu. V tem primeru bi bilo smiselno v množici vključiti zgolj prekrivanja, ki se tičejo našega problema, na primer prekrivanje oči zaradi las in očal, nas ne zanima. Naslednja metoda HPM se je bolje obnesla na našem problemu, četudi je dosegla na isti bazi primerljivo natančnost kot metoda RCPR [43]. Model, ki je bil vključen zraven naslednje metode, zazna večje

število obraznih značilk in sicer 68.

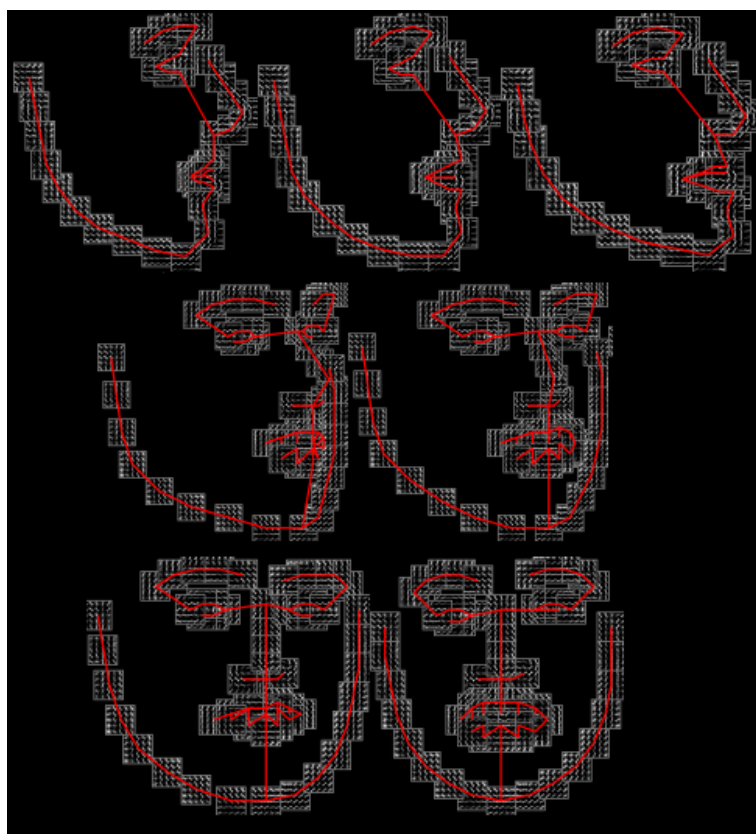
6.1.2 Hierarhični model iz več obraznih delov (HPM)

Opis metode

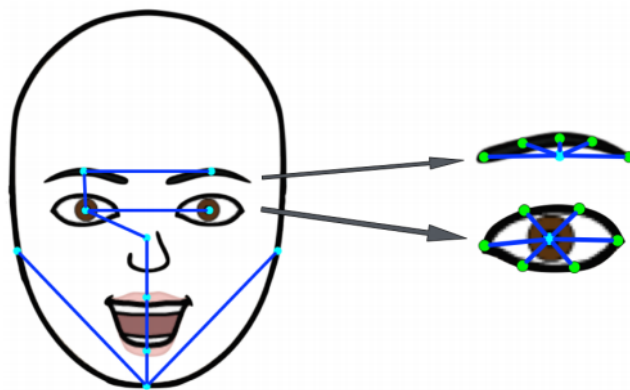
Metoda HPM (Hierarchical Part Model) [43] temelji na že obstoječemu detektorju [45], ki istočasno zazna obraz, obrazne značilke in smer pogleda obraza. Ta detektor obraz predstavi kot zmes dreves (ang. mixture of trees), vsaka obrazna značilka je predstavljena kot list drevesa, drevesna struktura se uporabi za ocenjevanje topoloških sprememb pri spremembi smeri pogleda. Vsaka izmed smeri pogleda, ki jo detektor lahko zazna, je predstavljena kot ločeno drevo. Prikaz tega je viden na sliki 6.7. Detektor na kateremu temelji metoda HPM je zmožen prepoznati tudi obraze s strani, torej obraz s smerjo pogleda -90 ali 90 stopinj. Za vsako izmed dreves se uporabi ocenjevalna funkcija, ki upošteva vizualni videz značilk, kot tudi postavitev obraznih značilk glede na omejitve oblike obraza. Detektor za vsako izmed smeri pogleda preišče sliko z drsečim oknom, slikovno piramido in linearnim klasifikatorjem. Ko detektor glede na vrednost ocenjevalne funkcije neko območje zazna kot obraz, tako dobimo zaznan obraz, skupaj z lokacijo obraznih značilk in oceno usmeritve pogleda.

Uporabljena implementacija metode HPM sicer zazna zgolj manjše število možnih smeri pogleda. Smer pogleda se zazna tremi modeli in sicer: levi model v območju -22.5 do -7.5 stopinj, središčni model v območju od -7.5 stopinj do 7.5 stopinj in desni model v območju od 7.5 do 22.5 stopinj. Na podoben način, kot izvorna metoda, metoda HPM predstavi obraz iz večih delov. Metoda HPM se razlikuje v tem, da uporabi prikaz sestavljen iz dveh hierarhij. Zgornja hierarhija je sestavljena iz delov obraza, kot so na primer oči, usta in obrvi, zgornja hierarhija je predstavljena v obliki drevesa. V spodnji hierarhiji so v obliki zvezde, obrazne značilke povezane v točko dela obraza na zgornji hierarhiji. Predstavitev zgornje in spodnje hierarhije je vidna na sliki 6.8.

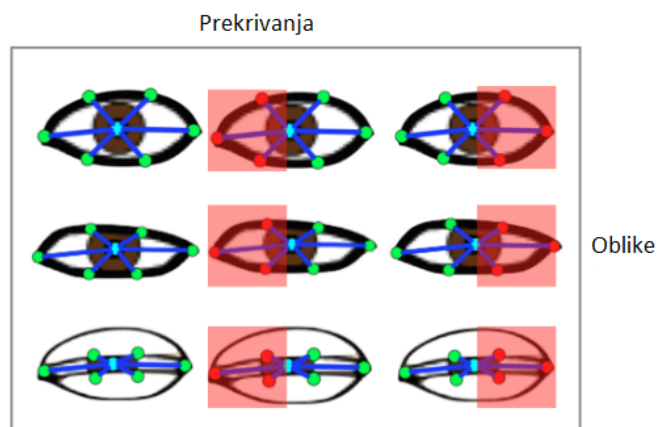
Za vsakega izmed obraznih delov in smeri pogleda, metoda HPM določi



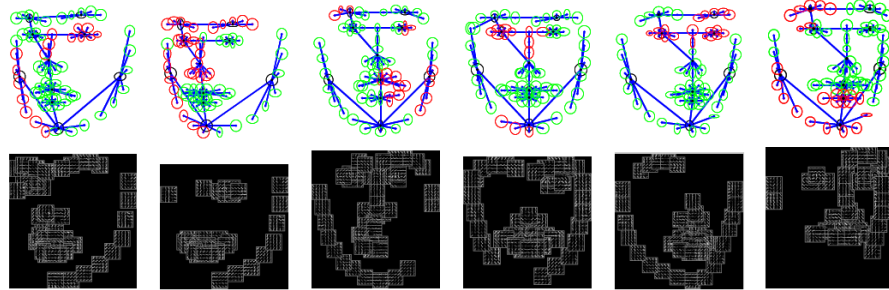
Slika 6.7: Prikaz predstavitve zmesi dreves za več možnih pogledov [45]. Rdeče črte predstavljajo drevo, ki povezuje obraze značilke. Obrazne značilke so vizualno predstavljene z značilkami HOG. Za učinkovitejše učenje si drevesa med seboj delijo nekatere izmed vizualnih predstavitev obraznih značilk.



Slika 6.8: Prikaz dveh nivojev hierarhije metode HPM. Na levi vidimo drevo obraznih delov, na desni pa kako se obrazne značilke povežejo v obrazni del [44].



Slika 6.9: Metoda HPM z uporabo različnih oblik obraznega dela in vzorcev prekrivanja generira veliko možnih primerov prekrivanja za posamezen del obraza [44].



Slika 6.10: Primeri različnih dreves, ki jih metoda HPM generira, glede na možne konfiguracije oblik obraznega dela in vzorcev prekrivanja obraznih značilk [43]. V spodnji vrsti vidimo, da je videz obraznih značilk predstavljen s HOG vrednostjo območja, prekrita območja imajo vrednost postavljeno na 0 (črna barva na sliki območja).

manjše število možnih zmesi, ki predstavljajo različne možne oblike in prekrivanja, primer tega je viden na sliki 6.9. Oblika predstavlja oddaljenost obraznih značilk od lokacije točke obraznega dela. Prekrivanje je določeno za vsako obrazno značilko posebej. Tako lahko model metode HPM deluje na različnem številu možnih obraznih izrazov in prekrivanj. Zato da je model naučen z metodo HPM bil sposoben zaznati čimveč možnih prekrivanj, so na učnih slikah generirali tudi sintetične primere prekrivanj. Primeri različnih končnih dreves, glede na izbrane konfiguracije oblik in vzorcev prekrivanj so vidni na sliki 6.10.

Za določanje lokacije delov obraza in obraznih značilk se uporablja ocenjevalna funkcija prikazana v enačbi 6.1. Črka l v enačbi predstavlja lokacijo obrazne značilke, s zmes oblike in o zmes prekrivanja. Prva vsota za lokacijo obraznih značilk določi vizualno ustreznost značilke. Utež v izbrani obliki lokacije se pomnoži z vizualnimi značilkami lokacije. Kadar je značilka prekrita, je vrednost produkta enaka nič. Druga vsota določi ustreznost lokacije obraznega dela glede na lokacijo obraznega dela, ki je njegov starš. Prvi člen vsote pomnoži lokacijo obeh obraznih delov s težo deformacije med njima. Predpostavlja se, da je relativna lokacija obraznih delov neodvisna od

prekrivanja na obrazu. Drugi člen vsote daje dodatno prednost določenim kombinacijam delov obraza (ang. bias term), glede na predstavitve oblike in prekrivanja, ki sta izbrani za ta dva dela obraza.

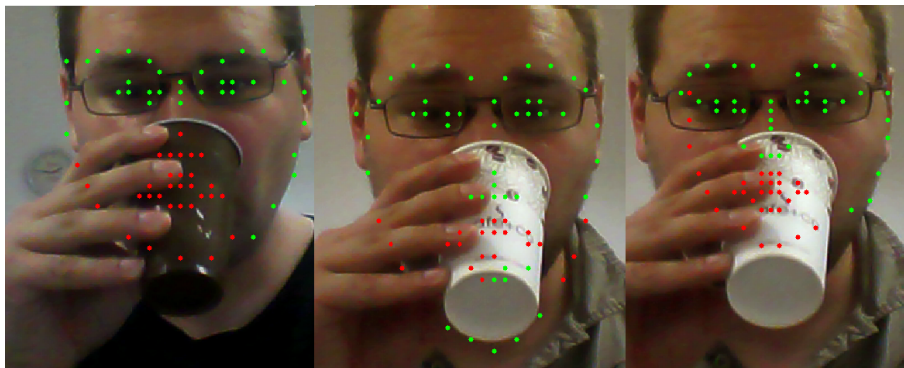
$$\begin{aligned}
 S(l, s, o) = & \sum_{i=obr.tocke} w_i(s_i) \cdot \theta_i(l_i, s_i) \\
 & + \sum_{i=deli\ obraza} \sum_{j=otroci(i)} (\vartheta_i(l_i, l_j) \cdot w_{ij}(s_i, s_j) \\
 & + bias(s_i, s_j, o_i, o_j)) \quad (6.1)
 \end{aligned}$$

Preizkus metode

Metoda je zelo počasna, zato smo jo preizkusili zgolj na naboru posnetkov iz spletne kamere. Posnetki so vsebovali različne scenarije prekrivanja recimo prekrivanje z lončkom, prekrivanje z roko in prekrivanje zaradi uporabe vilice. Metoda potrebuje v povprečju okoli 90 sekund za en posnetek iz spletne kamere, posnetki iz spletne kamere imajo razmerje slike 640×480 . Posnetki iz baze imajo namreč veliko večje razmerje slike, kar bi pomenilo še počasnejše delovanje.

Najboljše rezultate je metoda dala pri prekrivanju z lončkom. Med prekrivanjem so bile, izmed obraznih značilk ust in značilk okoli spodnjega roba obraza, zmeraj vsaj nekatere zaznane kot prekrite. Metoda je edino malce prezgodaj kazala konec prekrivanja, saj ko je bil obraz zgolj na dnu prekrit z lončkom, metoda praviloma ni zaznala prekrivanja. To težavo bi lahko rešili tako, da bi upoštevali obrazne sličice šele nekaj sličic po tem, ko ni bilo zaznano prekrivanje. Metoda bi bila torej za ta scenarij primerna. Metoda je sicer imela kar precej težav, pri ocenjevanju lokacije obraznih značilk ob prekrivanju z lončkom. Značilke so se ob prekrivanju pogostokrat določile precej više ali nižje od njihove prave lege. Pri pitju, ko je bil lonček najvišje postavljen, metoda večinoma ni bila sposobna zaznati obraza. Primeri prekrivanja z lončkom so vidni na sliki 6.11.

Metoda je tudi pri prekrivanju z roko dala dokaj solidne rezultate. Na-

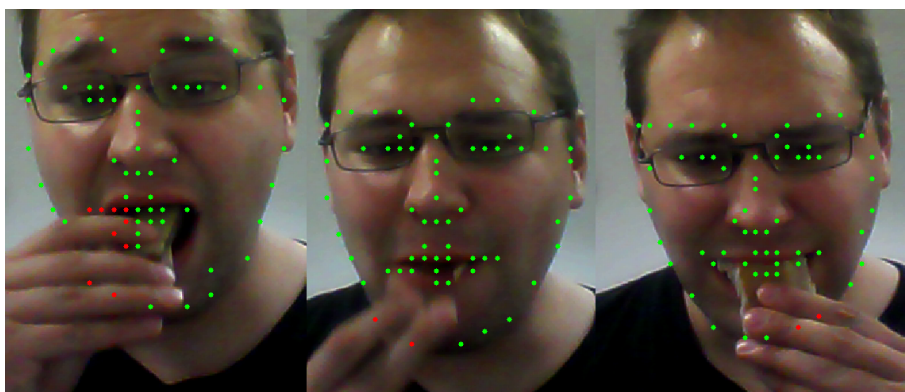


Slika 6.11: Primeri napovedovanja prekrivanja z lončkom z metodo HPM, na drugi in tretji sliki lahko vidimo, kako se je lokacija nekaterih obraznih značilk zaradi prekrivanja napačno ocenila.

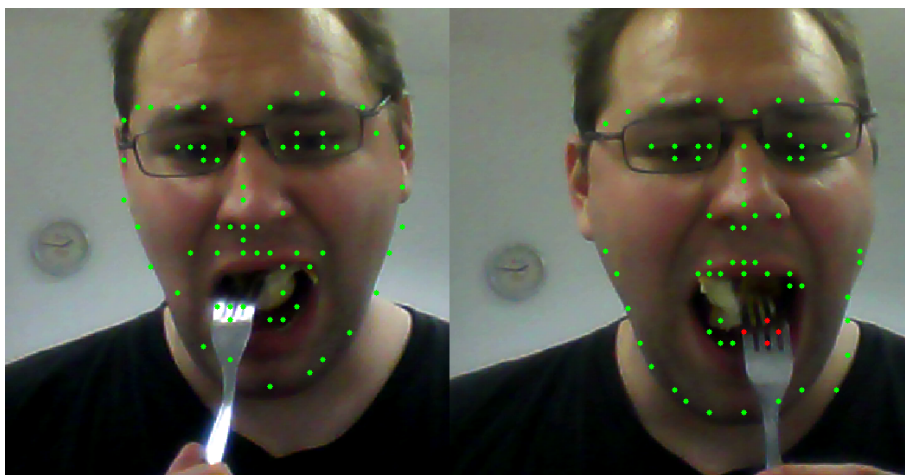
pram prekrivanju zaradi uporabe lončka je bil tukaj kot prekrit napovedan manjši delež obraznih značilk. V nekaterih primerih se je zgodilo, da na sliki s prekrivanjem, ni bila nobena obrazna značilka označena kot prekrita, kljub temu da so kasnejše slike imele pravilno zaznano prekrivanje značilk. To je dodaten razlog, da bi morali pri metodi konec prekrivanja določiti šele, kadar v nekaj zaporednih obraznih sličicah ni bilo zaznano prekrivanje. Primeri prekrivanja z roko so vidni na sliki 6.12.

Pri prekrivanju zaradi uporabe vilice, metoda razen v enem primeru ni zaznala prekrivanja vilice na spodnjemu robu obraza, nekajkrat je zaznala zgolj prekrivanje na obraznih značilkah v območju ust, večinoma zaradi hrane na vilici. Pri vseh posnetkih zajema hrane z vilico, roka ni prekrivala obraza, tako da je bilo območje prekrivanja majhno. Metoda je tako neprimerna za rabo, kadar prekrivanje povzroči ozek objekt. Primeri prekrivanja z vilico so vidni na sliki 6.13.

V vseh posnetkih skoraj ni bilo primerov, ko bi metoda označila neprekrите obrazne značilke kot prekrите, težave bi lahko nastopile, če bi metoda pogosto zaznala prekrivanje v območju ust, zaradi premikanja ust ob žvečenju hrane. Glavna težava metode je njena počasnost, v povprečju smo potrebovali okoli 90 sekund, za posamezno sličico, tako da je metoda popolnoma neprimerna



Slika 6.12: Primeri pravilno ugotovljenega prekrivanja zaradi roke z metodo HPM. Manjše število obraznih značilk je označeno kot prekritih, kot pa pri lončku. Metoda povečini dobro najde prekrivanje vendar ga na nekaterih sličicah ne zazna.



Slika 6.13: Ugotavljanje prekrivanja zaradi vilice z metodo HPM ne deluje dobro. V primerih ko je bilo prekrivanje zaznano, je bilo to v večini primerov okoli obraznih značilk ust in ne okoli spodnjega roba obraza.

za delo v realnem času.

6.2 Iskanje z detekcijo sprememb v vektorskem polju slike zaradi nastopa prekrivanja

6.2.1 Opis metode

Metoda se uporablja za ugotavljanje ali roka prekriva obraz [38]. Uporabljena je bila pri detekciji kretenj rok, razvita je bila tako, da zazna premike rok čez sliko obraza zaradi sprememb v sliki in istočasno ne daje napačnih napovedih pri spremembah zaradi premikov obraza ali spremembah v obraznem izrazu.

Metoda temelji na konceptu vektorskega polja slike (ang. image force field). Vektorsko polje slike opiše vsak slikovni element kot kombinacijo vseh ostalih slikovnih elementov slike. Vsak slikovni element vpliva na vse ostale slikovne elemente, sorazmerno glede na vrednost njegove intenzitete in obratno sorazmerno glede na razdaljo med slikovnima elementoma. Razdalja je definirana kot vektor dveh elementov, razdalje po širini in dolžini, glede na postavitev slikovnih elementov sta obe razdalji lahko negativni ali pozitivni. Enačba 6.2 prikazuje kako za posamezni slikovni element izračunamo vektorsko polje. Ta izračun se ponovi za vsak slikovni element, to predstavlja vektorsko polje slike. Razdalje med slikovnimi elementi so med trajanjem posnetka konstante, tako da jih izračunamo že pred potekom posnetka, vedeti moramo samo željeno velikost slike. Dimenzije vektorskega polja so enake dimenzijam slike, na katerih polje računamo.

$$FF(r_i, r_j) = \sum_{i=1, j=1}^{m, n} I(i, j) \frac{i - r_i, j - r_j}{|i - r_i, j - r_j|^3}, i! = r_i \vee j! = r_j \quad (6.2)$$

Potem ko smo za celotno sliko izračunali vektorsko polje slike, moramo izračunati usmerjenost vektorskega polja slike. V ta namen uporabimo funkcijo $\text{atan2}(y/x)$, y predstavlja imaginarno komponento vektorskega polja

slike, x pa realno komponento. Tako dobimo matriko kotov v dosegu $[-\pi, \pi]$. Spremembe v poteku usmerjenosti kotov so drugačne kadar roka prekriva obraz. Da te spremembe zaznamo matriko kotov konvolviramo s sobelovim operatorjem. Matriko ločeno konvolviramo z sobelovim jedrom po x in s sobelovim jedrom po y , obe matriki kvadriramo, seštejemo in nato vstoto matrik korenimo. Tako dobimo matriko sprememb kotov vektorskega polja. Na sliki 6.14 je prikaz matrik pred nastopom prekrivanja z roko in med prekrivanjem z roko. Matriko sprememb razdelimo v več regij in na vsaki regiji izračunamo vzorec LPB. Med potekom posnetka potem primerjamo vzorce LBP posameznih regij s prej izračunanimi vzorci LBP v prvi sliki. Za prvo sliko posnetka se smatra, da na obrazu ni prekrivanja, ni izrazitega obraznega izraza in da je obraz poravnan. Če spremembe na neki regiji presegajo določeno mejo, potem to območje označimo kot prekruto z roko. Primer iz članka metode je prikazan na sliki 6.15.

Uporabljena implementacija je s psevdoko prikazana v 3. Poleg uporabljene implementacije so spremembe v vektorskemu polju slike za detekcijo

ali roke prekrivajo obraz uporabili tudi v [39] in [40].

Algorithm 3: Ugotavljanje prekrivanja obraza z detekcijo sprememb
v vektorskemu polju

Podano imamo: parameter *meja_za_prekrivanje*, obmocja kjer
racunamo LBP in matriki vektorske_razdalje_x in vektorske_razdalje_y.
Ti dve matriki izracunamo lahko že pred začetkom algoritma, po
drugem delu produkta v enačbi 6.2, saj imajo vse slike zaporedja
enako velikosti. I je trenutna slika v zaporedju slik obrazov.

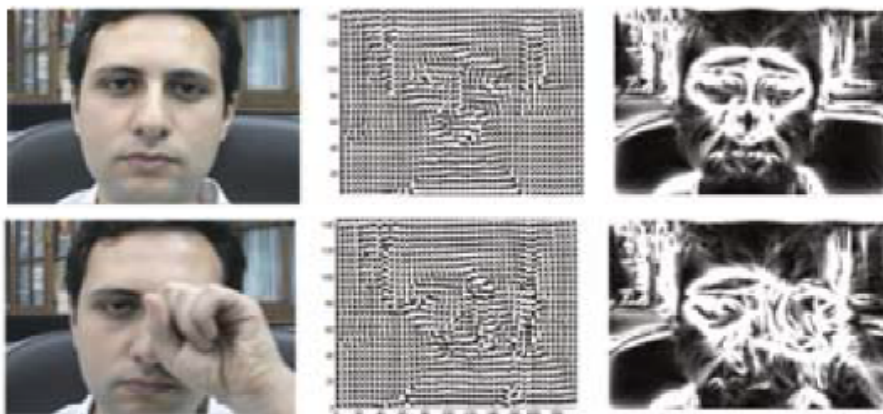
```

while nismo obdelali celotnega zaporedja slik obrazov do
    // izracunamo vektorsko_polje_x in vektorsko_polje_y, ki imata
    enako dimenzijo kot I M×N
    for a=1:M do
        for b=1:N do
            vektorsko_polje_x(a,b) =  $\sum(I \bullet \text{vektorske\_razdalje\_x}(a,b))$ 
            vektorsko_polje_y(a,b) =  $\sum(I \bullet \text{vektorske\_razdalje\_y}(a,b))$ 
        matrika_kotov = atan2(vektorsko_polje_y, vektorsko_polje_x)

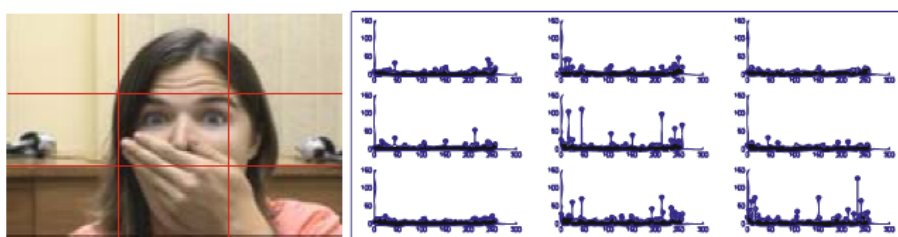
        sobelovo_jedro =  $\begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix}$ 

        H = konvolucija(matrika_kotov, sobelovo_jedro)
        V = konvolucija(matrika_kotov, sobelovo_jedroT)
        matrika_sprememb_kotov =  $\sqrt{H^2 + V^2}$ 
        for obmocje=1:stevilo_obmocij do
            lbp_hist = lbp(matrika_sprememb_kotov(obmocje))
            if I je prva slicica then
                | prvi_lbp_hist(obmocje) = lbp_hist
            else
                | razlika_histogramov =  $\sum(\text{abs}(\text{prvi\_lbp\_hist}(\text{obmocje}) -$ 
                | lbp_hist))
                if razlika_histogramov > meja_za_prekrivanje then
                    | v obmocju je prekrivanje

```



Slika 6.14: Prikaz sprememb, ki nastopijo ob prekrivanju z roko, kot so jih prikazali avtorji metode v članku [38]. Prvi stolpec vhodna slika, drugi stolpec matrika kotov, tretji stolpec matrika, ki jo dobimo z uporabo Sobelovega operatorja.



Slika 6.15: Spremembe v histogramu vzorca LBP po regijah, kot so jih avtorji metode prikazali v članku [38]. Vzorec LBP je bil izračunan na matriki sprememb kotov vektorskega polja. Spremembe so večje v regijah kjer je prisotna roka.

6.2.2 Rezultati metode

Iz naše baze posnetkov smo izbrali 31 posnetkov iz množice okušanja hrane, ki prikazujejo prekrivanje z roko. Posnetke smo spremenili v zaporedje sličic z uporabo detektorja obraza in obraznih značilnk iz knjižnice dlib. Glede na zaznana centra oči smo izvedli tudi poravnavo. Pri poravnavi smo izbrali manjšo velikost končne slike 100, za relativna centra levega očesa smo vzeli 0.28 za x in 0.20 za y , torej smo pustili na končni sliki večji del obraza, kot pa pri klasifikaciji. Za vsak posnetek smo pustili po koncu prekrivanja še dobrih 50 sličic. Na temu delu posnetka smo potem ugotavljali nastop in konec prekrivanja. Naša implementacija metode v Matlabu obdela v povpečju okoli pet sličic na minuto, kar je za rabo v realnem času prepočasno.

Določili smo tri regije kjer smo računali LBP vzorec, po višini vse tri regije potekajo v spodnji polovici slike. Prva regija poteka od začetka slike do polovice širine slike, druga regija poteka od četrte širine do tretje četrte širine, tretja regija poteka od polovice širine do konca slike obraza. Glede na parametre smo določili, kdaj nastopi prekrivanje v zaporedju slik in kdaj po nastopu prekrivanja začnemo upoštevati sličice za klasifikacijo. Preizkusili smo več možnih kombinacij parametrov in izbrali najboljšo. Nastop prekrivanja smo določili tako, da je vsota razlik LBP histograma bila vsaj 0.38, v vsaj eni regiji. V enem posnetku je bilo prekrivanje prej določeno, v treh posnetkih nobena izmed regij ni nikoli presegala meje 0.38, torej prekrivanje sploh ni bilo zaznano. V ostalih posnetkih je prekrivanje bilo določeno v pravem območju.

Nato smo pogledali koliko naslednjih sličic še ima vsaj eno regijo, ki presega mejo 0.38. Če bi začetek določili kot prvo sličico, ko nobena izmed regij ne presega več meje, bi se večina posnetkov začela na sličicah s prekrivanjem. Tako smo morali določiti dodatno število zaporednih sličic, ki jih bomo še izpustili. Ker v posnetkih odziv na okus hrane na obrazu nastopi šele po nekaj sličicah po prekrivanju in ker je večina posnetkov okušanja hrane dolgih, ni težav tudi če nekaj sličic po prekrivanju ne upoštevamo pri klasifikaciji. Kot najmanjše število izpuščenih sličic, kjer nobeden iz posnetkov nima več

prekrivanja, smo določili 22 sličic.

Metoda je bila v osnovi razvita za detekcijo prekrivanja obraza z roko, zato smo se odločili, da jo bomo tudi stestirali na posnetkih drugih vrst prekrivanja. Stestirali smo jo na posnetkih prek spletne kamere, katere smo uporabili prej pri prekrivanju z detektorjem obraznih značilk. Uporabili smo pet posnetkov prekrivanja z lončkom in pet posnetkov prekrivanja z vilico, obe vrsti posnetkov smo obravnavali ločeno. Pri posnetkih z lončkom smo kot idealno mejo našli vrednost 0.40. V vseh petih posnetkih smo s tem parametrom prav zaznali območje prekrivanja. Število dodatnih sličic, ki jih ne bomo več upoštevali, je pri prekrivanju z lončkom težje določiti. Detektor obraza iz dlib v nekaterih posnetkih zazna zelo malo sličic obraza, kadar ga prekriva lonček. Poleg tega so pri pitju pijače posnetki krajši, zato je treba biti bolj previden, da se ne izloči sličice kjer se kaže odziv na okus pijače. Za najboljšo oceno števila dodatnih izpuščenih sličic, bi bilo potrebno tudi upoštevati število sličic, kjer detektor obraza ni našel. Metoda je pravilno ocenila območje prekrivanja tudi pri prekrivanju pri zajemu hrane z vilico, kot mejo razlik smo spet določili 0.40. V vseh petih posnetkih je bil sicer na vilici dokaj velik kos hrane in je zato morala oseba pri zajemu hrane široko odpreti usta. Metoda sicer ni zaznala prekrivanja potem, ko je oseba pogoltnila hrano in je obraz prekrivala zgolj vilica, tako da je bilo prekrivanje zaznano ali zaradi hrane na vilici ali pa zaradi odpiranja ust.

Metoda načeloma deluje, vendar bi bilo potrebno še dodatno pretestirati izbiro idealnih parametrov. V nekaterih sličicah s prekrivanjem, je roka zgolj deloma vidna in zato je bila razlika LBP histograma ustrezne regije majhna. Zato bi bilo smiselno pretestirati parametre relativne lokacije centra oči, ki bi ohranili še večji del obraza in ozadja okoli obraza. Metoda je tudi občutljiva na prevelike spremembe v sliki pred nastopom prekrivanja zaradi zajema hrane.

V tem poglavju smo predstavili pristope za odkrivanje prekrivanja, ki nastane pri zajemu hrane. Od dveh testiranih obraznih detektorjev obraznih značilk s prekrivanjem, se je eden pokazal kot dovolj natančen, vendar pa tudi kot prepočasen za dejansko rabo. Detektor obraznih značilk, ki zazna

prekrivanja, bi bilo smiselno trenirati na množici, ki ima okoli roba obraza in ust večje število obraznih značilk in vsebuje zgolj prekrivanja, ki nastanejo pri zajemu hrane. Ugotavljanje prekrivanja glede na spremembe v vektorskem polju slike lahko zazna prekrivanje, vendar pa je zelo občutljivo glede na izbiro parametrov, tukaj bi bilo potrebno še veliko dodatnega testiranja.

Poglavje 7

Sklepne ugotovitve

V magistrski nalogi smo prikazali uporabo metod za prepoznavo obraznega izraza za napovedovanje ali je osebi okus zaužite hrane všeč. Potek opravljenega dela lahko razdelimo v tri sklope. V prvem sklopu smo metodo preizkusili na standardnih bazah, kjer je predstavljen izraz več različnih čustev. Metodo smo stestirali na bazah Cohn Kanade CK+ in GEMEP-FERA, kjer smo dosegli zelo dobre rezultate. Na bazi GEMEP-FERA smo dosegli 79.85% natančnost. Na naši delitvi baze Cohn Kanade CK+ na učno in testno množico smo dosegli 96.42% natančnost. Metodo smo nato izboljšali tako, da je bila primerna za delo v realnem času na lastnih posnetkih. V drugem sklopu smo metodo preizkusili na problemu ugotavljanja prijetnosti okusa zaužite hrane. Problem smo predstavili kot dvorazredni klasifikacijski, bazo posnetkov smo pripravili sami. Baza je sestavljena iz množice posnetkov pitja pijače in množice posnetkov okušanja hrane. Natančnost metode na obeh množicah smo ovrednotili z večkratnim prečnim preverjanjem. Dobili smo zelo dobre rezultate na množici pitja pijače in dobre rezultate na množici okušanja hrane. Pri pijači je povprečna natančnost 10 prečnih preverjanj 92%, pri hrani je povprečna natančnost 10 prečnih preverjanj 81%. Podrobno smo tudi analizirali napake. Ugotovili smo, da se večina napak pojavi v posnetkih okušanja hrane, kadar posnetek predstavlja neprijeten odziv na okus hrane, vendar na obrazu okuševalca neprijetni odziv ni viden. V tretjem

sklopu smo obravnavali metode za odkrivanje prekrivanja obraza, ki nastane pri zaužitju hrane. Namesto specifičnih metod, ki so odvisne od vrste prekrivanja, smo preizkusili čimbolj splošne metode. Preizkusili smo detektorje obraznih značilk, ki poleg lokacije obraznih značilk tudi zaznajo, ali so obrazne značilke prekrite in detekcijo prekrivanja obraza glede na spremembe v vektorskemu polju slike. Ugotovili smo, da sta pristopa v osnovi primerna za naš problem, vendar bi bilo za dejansko rabo potrebno še dodatno izboljšati njuno natančnost in hitrost delovanja.

Poleg že omenjenih izboljšav metod za odkrivanje prekrivanja so možne še naslednje izboljšave. Baza trenutno vsebuje zgolj pet oseb, kar je majhno število, tako da bi bilo bazo potrebno dopolniti s posnetki dodatnih oseb. V primeru več oseb bi lahko bazo ovrednotili na videnem in nevidnem delu množice - osebah ki niso nastopale v učni množici, torej podobno kot pri bazi GEMEP-FERA. Trenutno se uporablja zelo osnovna metoda za poravnavo obraza, katero bi lahko nadomestili z naprednejšo. V magistrski nalogi smo na bazi uporabili zgolj eno vrsto značilk in sicer LBP. Kot druge primerne značilke bi lahko uporabili LPQ, HOG ali pa HOG z uporabo piramide - PHOG (Pyramid Histogram of Oriented Gradients). Lahko bi uporabili tudi razširitve LBP in LPQ s treh ortogonalnih ravnin (ang. Three Orthogonal Planes) LBP-TOP in LPQ-TOP. Ti dve predstavitvi upoštevata tudi časovne spremembe vzorca, vendar imata veliko večjo dimenzionalnost kot osnovni verziji teh značilk.

Literatura

- [1] Y. Tian, T. Kanade, J. Cohn, “Facial Expression Analysis”, 11. poglavje v Handbook of Face Recognition. ZDA: Springer, 2011
- [2] Y. Tong, W. Liao, Q. Ji, “Inferring Facial Action Unit with Causal Relations”, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2006*, zv. 2, str. 1623-1630, 2006
- [3] “Facial Expressions and Emotions”, dostopno na: <https://writersforensicsblog.wordpress.com/2013/08/19/facial-expressions-and-emotions/>
- [4] P. Viola, M. Jones, “Robust Real-Time Face Detection”, *International Journal of Computer Vision*, št. 57, zv. 2, str. 137–154, 2004.
- [5] T. Ojala, M. Pietikainen, T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, št. 7, zv. 24, str. 971–987, IEEE, 2002.
- [6] M. Pietikainen, J. Heikkila, “Image and Video Description with Local Binary Pattern Variants Tutorial”, dostopno na: <http://www.ee.oulu.fi/research/imag/mvg/files/pdf/CVPR-tutorial-final.pdf>
- [7] X. Yan, “Constrained Local Model for Face Alignment, a Tutorial ”, Dostopno na: <https://781d5d26-a-62cb3a1a-s-sites.googlegroup.s.com/site/xgyanhome/home/projects/clm-implementation/ConstrainedLocalModel-tutorial%2Cv0.7.pdf>

-
- [8] “Introduction to Support Vector Machines”, Dostopno na: http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html
 - [9] E. Kim, “Everything You Wanted to Know about the Kernel Trick (But Were Too Afraid to Ask)”, Dostopno na: http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html
 - [10] S. Emani, “Face Recognition using Eigenfaces or Fisherfaces”, 8. poglavje v *Mastering OpenCV with Practical Computer Vision Projects*. ZDA: Packt Publishing, 2012
 - [11] G. Borenstein, “Slika haar cascade frontal face alt”, dostopno na: <https://www.flickr.com/photos/unavoidablegrain/6884354620/in/photostream/>
 - [12] C. de Souza, “Haar-feature Object Detection in Csharp”, dostopno na: <http://www.codeproject.com/Articles/441226/Haar-feature-Object-Detection-in-Csharp>
 - [13] “Local Representation of Facial Features (Face Image Modeling and Representation) (Face Recognition) Part 1”, dostopno na: <http://what-when-how.com/face-recognition/local-representation-of-facial-features-face-image-modeling-and-representation-face-recognition-part-1/>
 - [14] S. Yang, B. Jiang, “Understanding discrete facial expressions in video using an emotion avatar”, *International Journal of Computer Vision*, št. 4, zv. 42, str. 980-992, 2012.
 - [15] J. Saragih, S. Lucey, J. Cohn, “Face alignment through subspace constrained mean-shifts”, *IEEE international conference on computer vision (ICCV 09)*, str. 1034–1041, IEEE, 2009.

-
- [16] D. Comaniciu, P. Meer, "Mean shift: A robust approach toward feature space analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, št. 5, zv. 24, str. 603-619, 2002.
- [17] K. Wendin, BH. Allesen-Holm, WLP. Bredie, "Do facial reactions add new dimensions to measuring sensory responses to basic tastes?", *Food Quality and Preference*, št. 4, zv. 22, str. 346-354, 2011.
- [18] K. Sjøgren, "Your face reveals what you are eating", dostopno na: <http://sciencenordic.com/your-face-reveals-what-you-are-eating>
- [19] G. Zeinstra, "Children's food preferences: Facial expression analysis with The Observer XT", dostopno na: http://www.noldus.com/files/file_manager/downloads/case_studies/Children's%20food%20preferences%20-%20facial%20expression%20analysis%20with%20The%20observer%20XT.pdf
- [20] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, str. 94-101, IEEE, 2010
- [21] "Noldus Face Reader", dostopno na: <http://www.noldus.com/human-behavior-research/products/facereader>
- [22] "New Release: FaceReader5.0 – software for automatic facial expression analysis", dostopno na: <http://www.prlog.org/12033160-new-release-facereader50-software-for-automatic-facial-expression-analysis.html>
- [23] "Odprtokodna knjižnica OpenCV", dostopno na: <http://opencv.org/about.html>
- [24] D. E. King "Dlib-ml: A Machine Learning Toolkit", *Journal of Machine Learning Research*, št. 10, str. 1755-1758, 2009

-
- [25] “Dlib 18.6 released: Make your own object detector!”, dostopno na: <http://blog.dlib.net/2014/02/dlib-186-released-make-your-own-object.html>
- [26] V. Kazemi, J. Sullivan, “One Millisecond Face Alignment with an Ensemble of Regression Trees”, *CVPR '14 Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition* str. 1867 - 1874, IEEE, 2014
- [27] M. F. Valstar, M. Mehu, B. Jiang, M. Pantic, K. Scherer, “Meta-analysis of the first facial expression recognition challenge”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, št. 4, zv. 42, str. 966-979, 2012
- [28] N. Dalal, B. Triggs, “Histograms of oriented gradients for human detection”, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, zv. 1, str. 886-893, 2005
- [29] C. Liu, J. Yuen, A. Torralba, “Sift flow: Dense correspondence across scenes and its applications”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, št. 5, zv. 33, str. 978-994, 2011
- [30] E. Rahtu, J. Heikkilä, V. Ojansivu, T. Ahonen, “Local phase quantization for blur-insensitive image analysis”, v reviji *Image and Vision Computing*, št. 8, zv. 30, str. 501-512, Elsevier, 2012
- [31] P. Lucey, J. Cohn, K. Prkachin, P. Solomon, I. Matthews, “Painful data: The UNBC-McMaster shoulder pain expression archive database”, *IEEE International Conference on Automatic Face and Gesture Recognition and Workshops (FG 2011)*, str. 57-64, IEEE, 2011
- [32] L. Zhang, D. Tjondronegoro, V. Chandran, “Representation of facial expression categories in continuous arousal-valence space: Feature and correlation”, *Image and Vision Computing*, št. 12, zv. 32 str. 1067-1079, IEEE, 2011

-
- [33] L. Danner, L. Sidorkina, M. Joechl, K. Duerrschmid, "Make a face! Implicit and explicit measurement of facial expressions elicited by orange juices using face reading technology", v zborniku *Food Quality and Preference*, zv. 32, str. 167-172, 2012
- [34] D. Garcia-Burgos, MC. Zamora, "Facial affective reactions to bitter-tasting foods and body mass index in adults", *Appetite*, zv. 71, str. 178-186, Elsevier, 2013
- [35] RA. de Wijk, V. Kooijmana, RHG. Verhoevenb, NTE. Holthuysena, C. de Graafc "Autonomic nervous system responses on and facial expressions to the sight, smell, and taste of liked and disliked foods", *Food Quality and Preference*, št. 2 zv. 26, str. 196-203, 2012
- [36] RA. de Wijk, V. Kooijmana, RHG. Verhoevenb, NTE. Holthuysena, C. de Graafc "ANS Responses and Facial Expressions Differentiate between the Taste of Commercial Breakfast Drinks", *PLoS ONE*, 2014
- [37] G. Juodeikiene, L. Basinskiene, D. Vidmantienė, D. Klupsaite, E. Bartkiene, "The use of face reading technology to predict consumer acceptance of confectionery products", *9th Baltic Conference on Food Science and Technology*, 2014
- [38] M. Mahmoud, R. El-Kaliouby, A. Goneid, "Towards Communicative Face Occlusions: Machine Detection of Hand-over-Face Gestures", *ICIAR '09 Proceedings of the 6th International Conference on Image Analysis and Recognition*, str. 481-490, Springer, 2009
- [39] P. Smith, N. da Vitoria Lobo, M. Shah, "Towards Communicative Face Occlusions: Machine Detection of Hand-over-Face Gestures", *Image and Vision Computing*, št. 9, zv. 25, str. 481-490, Elsevier, 2007
- [40] J. Xu and X. Zhang, "A Real-Time Hand Detection System during Hand over Face Occlusion", *International Journal of Multimedia and Ubiquitous Engineering*, št. 8, zv. 10, str. 481-490, SERSC, 2015

-
- [41] X. P. Burgos-Artizzu, P. Perona and P. Dollar, “Robust face landmark estimation under occlusion”, *IEEE International Conference on Computer Vision (ICCV)*, 2013, str. 1513-1520, IEEE, 2013
 - [42] X. P. Burgos-Artizzu, P. Perona and P. Dollar, “Robust face landmark estimation under occlusion”, poster, dostopno na: <http://www.vision.caltech.edu/xpburgos/ICCV13/Data/Poster.pdf>
 - [43] G. Ghiasi, C. Fowlkes, “Occlusion Coherence: Localizing Occluded Faces with a Hierarchical Deformable Part Model”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014*, str. 1899-1906, IEEE, 2014
 - [44] G. Ghiasi, C. Fowlkes, “Occlusion Coherence: Localizing Occluded Faces with a Hierarchical Deformable Part Model”, poster, dostopno na: <https://www.ics.uci.edu/~gghiasi/papers/gf-cvpr14-poster.pdf>
 - [45] X. Zhu, D. Ramanan, “Detection, Pose Estimation, and Landmark Localization in the Wild”, *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, str. 2879-2886, IEEE , 2012